Version 5.0
January, 2002

HST Data Handbook for NICMOS

SPACE
TELESCOPE
SCIENCE
INSTITUTE

Hubble Division
3700 San Martin Drive
Baltimore, Maryland 21218
help@stsci.edu

Operated by the Association of Universities for Research in Astronomy, Inc., for the National Aeronautics and Space Administration

User Support

For prompt answers to any question, please contact the STScI Help Desk.

- **E-mail:** help@stsci.edu
- **Phone:** (410) 338-1082
  (800) 544-8125 (U.S., toll free)

World Wide Web

Information and other resources are available on the Web site:

- **URL:** http://www.stsci.edu.

NICMOS Revision History

Version	Date	Editors
1.0	February 1994	Stefi Baum
2.0	December 1995	Claus Leitherer
3.0, Vol., I	October 1997	Mark Voit
3.0, Vol. II	October 1997	Tony Keyes
3.1	March 1998	Tony Keyes
4.0	December 1999	Mark Dickinson
5.0	January 2002	Bahram Mobasher, Chief Editor, HST Data Handbook Michael Corbin, Jin-chung Hsu, Editors, HST Introduction Mark Dickinson, Editor, NICMOS Data Handbook

Contributors

M. Dickinson, S. Arribas, L. E. Bergeron, T. Boeker, D. Calzetti, S. T. Holfeltz, B. Mobasher, B. Monroe, K. Noll, E. Roye, A. Schultz, M. Sosey, C. Xu.

Citation

In publications, refer to this document as:
Dickinson, M. E. et al. 2002, in HST NICMOS Data Handbook v. 5.0, ed. B. Mobasher, Baltimore, STScI

Send comments or corrections to:
Hubble Division
Space Telescope Science Institute
3700 San Martin Drive
Baltimore, Maryland 21218
E-mail: help@stsci.edu

Table of Contents

Part I: Introduction to Reducing HST Data

Part II: NICMOS Data Handbook

Chapter 4: Anomalies and Error Sources

Chapter 5: Data Analysis 5-1

Part III: Appendixes

Preface

Introduction to Reducing HST Data

This data handbook provides an introduction to the process of retrieving and reducing Hubble Space Telescope (HST) data. The reduction procedures, calibrations, and sources of error specific to each active HST instrument (WFPC2, STIS, NICMOS and ACS) are explained in detail in their respective instrument data handbooks. However, we recommend a careful reading of this handbook before proceeding to the instrument data handbooks and before starting to work on your HST data. The present document is an updated version of chapters 1-3 of the HST Data Handbook, v.3.0, and is based on the information available as of December 2001. In particular, it is written under the assumption that the ACS and NICMOS instruments will be fully functional following HST Servicing Mission 3B (SM3B).

Many changes in the HST Data Archive and HST data reduction software have occurred since v. 3.0 of the Hubble Data Handbook. These differences are covered in this document and include, but are not limited to:

1. Expansion of the HST Data Archive into the Multimission Archive at Space Telescope (MAST), which currently includes 14 satellite mission archives as well as ground-based survey data.
2. The ability to retrieve MAST data using the World Wide Web.
3. New capabilities of the StarView program for searching for and retrieving HST and other MAST data.
4. The introduction of PyRAF, a new Python-based alternative to the IRAF cl shell, and
5. A new distinction between waiver FITS format, used to archive data from the older HST instruments such as WFPC and FOS, and the FITS extension format used for the newest instruments (STIS, NICMOS and ACS).

Future changes in this handbook are anticipated as MAST expands to cover additional missions, and as StarView and PyRAF evolve. The reader is advised to consult the STScI web site at http://resources.stsci.edu for the latest information. Moreover, as the present revision comes before SM3B, important revisions to the ACS file structure and data handling may be necessary after the installation of this instrument.

Bahram Mobasher (Chief Editor, HST Data Handbook)
Michael Corbin (Editor, Chapter 1)
Jin-chung Hsu (Editor, Chapters 2 and 3)

PART I:

Introduction to Reducing HST Data

This part of the data handbook provides an introduction to the process of retrieving and reducing Hubble Space Telescope (HST) data.

CHAPTER 1:

Getting HST Data

In this chapter. . .

This chapter describes how to obtain Hubble Space Telescope (HST) data files. All HST data files are stored in the Hubble Data Archive (HDA), which forms part of the Multimission Archive at STScI (MAST)[1]. HST Guaranteed Time Observers (GTOs), Guest Observers (GOs) and Archival Researchers can retrieve data in either of two ways:

- Electronically over the Internet from the HDA, where data are stored immediately after they pass through HST pipeline processing.
- On data storage media written at STScI from the HDA. The options are Exabyte and DAT tapes, and will include CDs and DVDs in the future.

To retrieve data electronically you must first register as a MAST user[2]. HST Principal Investigators (PIs) are ***not*** automatically registered. If you have not recently retrieved data, you should register or renew your registration before retrieving data from the HDA. PIs should register before their observations are made. GTO and GO observations normally remain proprietary for a period of one year, which means that during this period

[1] MAST currently includes data from HST, FUSE, IUE, EUVE, ASTRO, HUT, UIT, WUPPE, ORFEUS, BEFS, IMAPS, TUES, Copernicus and ROSAT. Data from the FIRST radio survey, Digital Sky Survey (DSS) and Sloan Digital Sky Survey (SDSS) are also available.

[2] By 2002, registration will no longer be required for public (non-proprietary) data.

other registered users cannot retrieve them without authorization from the PI. All calibration observations as well as observations made as part of the Public Parallel programs are immediately public. All observations made as part of the Treasury Programs begun in Cycle 11 will either be immediately public or have only a brief proprietary period. The HST section of MAST also contains several Prepared (fully reduced) data sets, including the Hubble Deep Fields, the Hubble Medium Deep Survey Fields, and a composite quasar spectrum, which are also public.

This chapter describes how to search the HDA, how to electronically retrieve files from it, and how to request and read tapes and disks containing HST data. As an aid to retrieving their data, PIs will automatically receive e-mail notification of the status of their observations two times: first, when the first datasets for their proposal are archived, and second, when all the datasets for their proposal and all necessary calibration files have been archived.

Note for Advanced Camera for Surveys (ACS) Users: Calibrated ACS images are approximately 168 MB in size, larger than those of any other HST instrument. Therefore, electronic retrieval of ACS data is enabled only for those with broadband (> 100 KB/s) Internet connections, in order to ensure uninterrupted transmission of individual files. Users retrieving large numbers of ACS files should also consider requesting them on tape or disk.

1.1 Archive Overview

The HDA contains all HST observations ever made. It also contains a database that catalogs and describes these observations. There are currently two ways to search and retrieve data from the HDA. The first is a program called StarView, which acts as an interface to the HDA. StarView currently runs as Java-based, stand-alone application that can be downloaded from the web site http://starview.stsci.edu. Previous versions of StarView, such as XStarView, are no longer available. The second search and retrieval method is through the HST section of the MAST web site, http://archive.stsci.edu. StarView is the more powerful of the two methods, and in particular allows an examination of the calibration files applied to a given data file. StarView also provides an interface to the Visual Target Tool (VTT) in the Astronomer’s Proposal Tool (APT) suite of programs. The VTT interface can display archive observations on a Digital Sky Survey (DSS) image alongside planned observations. StarView is thus recommended for observation planning, duplication checking, calibration file review, investigation of On-The-Fly Reprocessing flags and proprietary status. It is also recommended for those needing to retrieve large numbers of datasets, and those needing to examine calibration files. The MAST web site interface to the HDA has the same basic capabilities as StarView, and may be preferable for those requiring simple retrievals of datasets. Both StarView and the MAST web site allow simultaneous searches of the other MAST mission archives for all HDA searches. They also offer simple preview of the capabilities of HST datasets when available, as well as links to literature references citing a given dataset, using the Astrophysics Data System (ADS). In later sections we discuss StarView and the MAST web site in more detail.

1.1.1 Archive Registration

The simplest way to register and retrieve HST data is to complete the form on the Web page at: http://archive.stsci.edu/registration.html.

Registration requests may also be sent to the HDA hotseat, at: archive@stsci.edu.

The PI of each HST proposal must request proprietary access for their data, and for anyone else whom the PI wants to have access to it. PI retrieval permission is not granted automatically, for security reasons. PIs wishing to allow others access to their proprietary data should make that request to archive@stsci.edu.

When registration is granted, your account will be activated within two working days, and you will receive your username and password via e-mail.

1.1.2 Archive Documentation and Help

The MAST web site provides a wealth of useful information, including an online version of the HST Archive Manual available at http://archive.stsci.edu/hst/manual/. Investigators expecting to work regularly with HST and other datasets supported by MAST should also subscribe to the MAST electronic newsletter by sending an e-mail to archive_news-request@stsci.edu and putting the single word *subscribe* in the body of the message. Questions about the HDA can be directed to archive@stsci.edu, or by phone to (410) 338-4547.

1.2 Getting Data with StarView

1.2.1 Downloading and Setting Up StarView

The latest version of StarView runs under versions 1.2.2 and later of Java and may be downloaded from http://starview.stsci.edu.

This site also includes a FAQ page and news on releases and updates. StarView will automatically update itself to the latest version, so users do not have to worry about additional installations. Following its installation on computers running Unix and Linux, begin StarView by typing

```
> StarView
```

at the system prompt. Under Windows and MacIntosh systems, StarView will appear as an icon. The StarView session then begins, first with an Information window explaining navigation within StarView, and a request for the user to specify an object name resolver (SIMBAD or NED) for use in HDA searches. The first-time user are asked to supply their e-mail information in order to allow StarView to communicate the results of its attempts to retrieve the files requested from the HDA. This e-mail information includes the user's SMTP host, or the computer from which e-mail messages are routed. If unsure of your SMTP host, ask your system administrator. These queries can be turned off for future sessions once this information has been supplied.

1.2.2 Simple Use of StarView

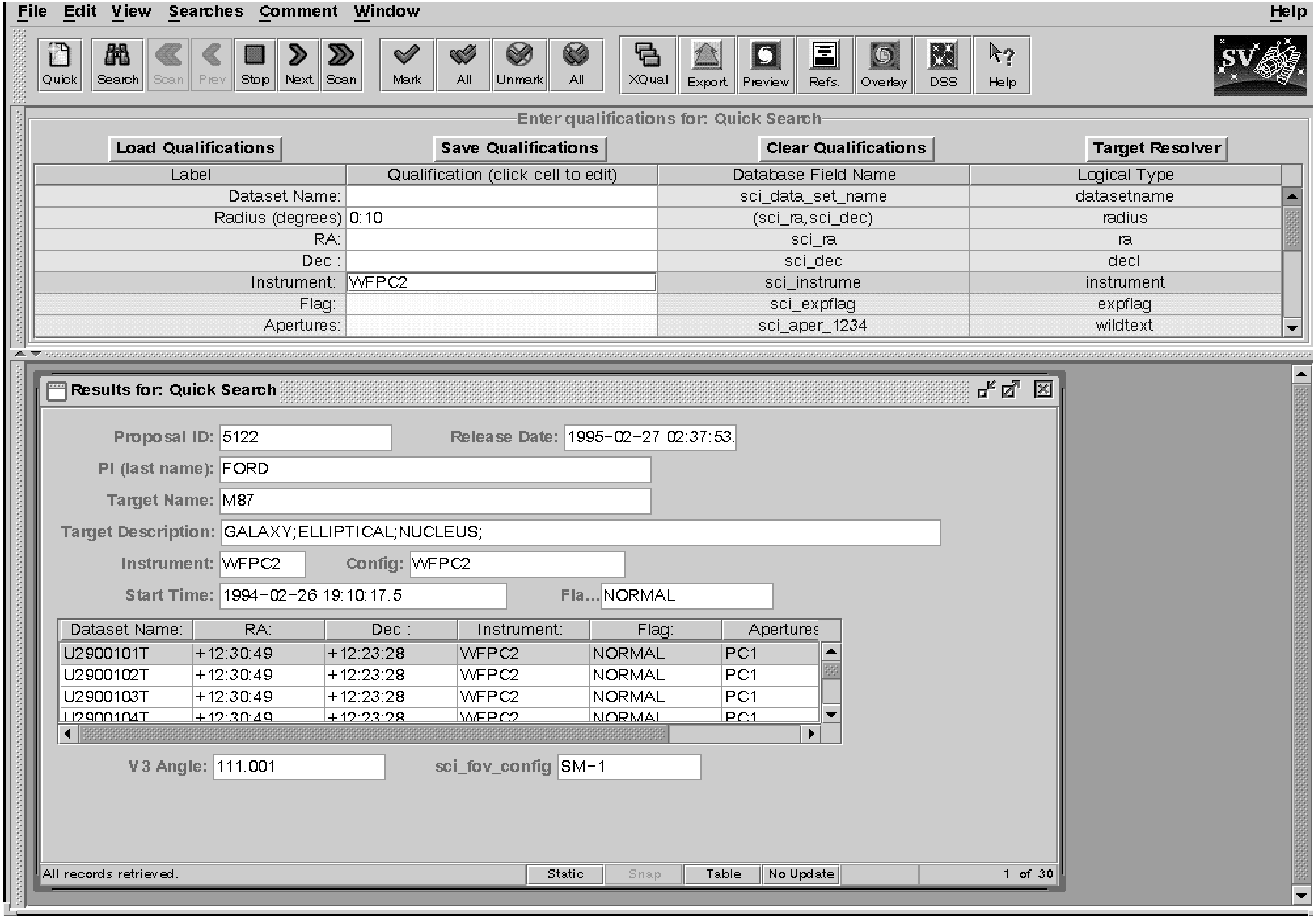
We now proceed to an introduction to the use of StarView. A more detailed description of its capabilities is provided at the web site above, which should also be consulted for more advanced topics such as its Table Exportation and Cross-Qualification functions.

The basic function of StarView is to enable the user to first search the HDA (and the other mission archives in MAST) for data files matching criteria such as object name, position, or proposal number, then allow the user to navigate through the set of files matching those criteria, and finally to let the user retrieve some or all of the files found in the search. Several options for the type of search that can be performed (e.g. by a particular instrument) will be discussed later.

The design of StarView is similar to that of a Web browser. At its top are pull-down menu bars including File, Searches, and Help. The Help menu offers links to documents including the StarView FAQ page. Beneath these menu bars is a row of buttons that run StarView's basic functions, such as searching, marking files for retrieval, and previewing images. A Help button allows users to display pop-up windows describing the function of the different StarView buttons and windows, by first clicking the Help button, then the item of interest. Beneath the row of buttons is the Qualifications table, which is displayed when a search is begun. It consists of several cells corresponding to the search parameters the user wishes to use, e.g., object name, proposal I.D., or instrument. Below this window will appear the Results table, displaying the datasets found to match a given set of search parameters entered into the Qualifications table. For the purpose of introduction, we will describe the use of the most basic search option, called "Quick Search," which can be started by clicking the "Quick" button at the top left of StarView.

As an example of the use of the Quick Search option, we will request all available WFPC2 data for the galaxy M87. This is done by typing "WFPC2" and "M87" in the Instrument and Target Name cells of the Qualifications section, then clicking the "Search" button at the top left of the StarView window. The results of the search will then be displayed in the bottom panel of StarView, as shown in figure 1.1. These results include the dataset name, instrument name, R.A. and Dec of the target, and the instrument aperture used. Note that these parameters could also have been specified in the Qualifications section, as can other parameters including proposal I.D. number, proposal P.I. name, and image central wavelength (corresponding to particular instrument filters or gratings).

Figure 1.1: Results of StarView Quick Search for WFPC2 files of M87



Enter qualifications for: Quick Search

Label	Qualification (click cell to edit)	Database Field Name	Logical Type
Dataset Name:		sci_data_set_name	datasetname
Radius (degrees)	0.10	(sci_ra,sci_dec)	radius
RA:		sci_ra	ra
Dec :		sci_dec	decl
Instrument:	WFPC2	sci_instrume	instrument
Flag:		sci_expflag	expflag
Apertures:		sci_aper_1234	wildtext

Results for: Quick Search

Proposal ID: 5122
Release Date: 1995-02-27 02:37:53.
PI (last name): FORD
Target Name: M87
Target Description: GALAXY;ELLIPTICAL;NUCLEUS;
Instrument: WFPC2
Config: WFPC2
Start Time: 1994-02-26 19:10:17.5
Fla... NORMAL

Dataset Name:	RA:	Dec :	Instrument:	Flag:	Apertures
U2900101T	+12:30:49	+12:23:28	WFPC2	NORMAL	PC1
U2900102T	+12:30:49	+12:23:28	WFPC2	NORMAL	PC1
U2900103T	+12:30:49	+12:23:28	WFPC2	NORMAL	PC1
U2900104T	+12:30:49	+12:23:28	WFPC2	NORMAL	PC1

V3 Angle: 111.001
sci_fov_config SM-1

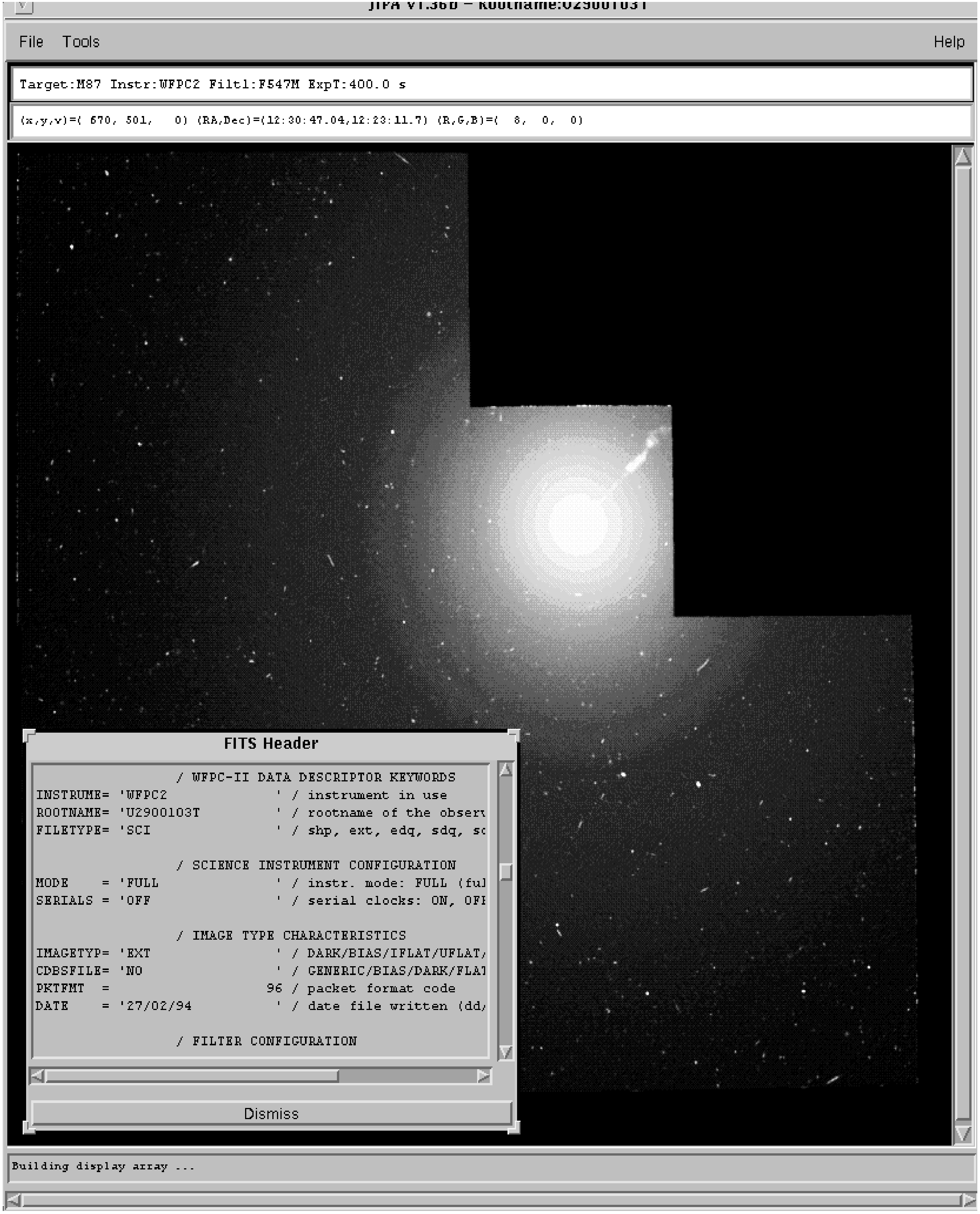
All records retrieved.
1 of 30

Clicking on a given dataset in the Results table will display the information shown in the cells above it (Proposal ID, Release Date, P.I., etc.). You may browse through the retrieved datasets either by using the mouse and scroll bar, or by using the navigation buttons (Scan, Previous, Next) in the top row of mouse buttons. The Scan option will automatically step through all of the files retrieved in the search, provided that the right most button at the bottom of the Results window is toggled to “Update.” If this button is toggled to “No Update,” the Scan option will go straight to the end of the list of files.

The ability to obtain a preview is available for many, but not all, of the datasets in the HDA (e.g., previews are not available for many FOC datasets). This is done with the “Preview” button, if it is enabled. For images, this will display a re-sampled version of the image using the Java Image Preview Application (JIPA) tool that is part of StarView. For spectra, a simple GIF image of the calibrated spectrum will be displayed. JIPA and VTT can also display an image's FITS header, under the JIPA “Tools” menu. The JIPA preview of the WFPC2 image U2900103T retrieved in the previous search for WFPC2 images of M87 is shown in figure 1.2, along with the window displaying part of the FITS header file of this image.

Other display options with StarView include “DSS,” which will display a 20' x 20' Digital Sky Survey image at the target coordinates, while the “Overlay” button will display the same DSS image with outlines of the HST instrument apertures at the target coordinates superimposed on it, at the orientation of the observation selected. The “References” button provides a link to any known published papers citing the dataset, as listed in ADS. Note that the HST images displayed by the Preview are of reduced quality compared to the actual data files, and cannot be downloaded. They are only meant to provide a quick check that the datasets found by the search met the search criteria, i.e., contained the object(s) of interest, and are of the desired quality.

Figure 1.2: JIPA preview of WFPC2 image U2900103T, along with image header file, using Preview option



```
Target:M87 Instr:WFPC2 Filt1:F547M ExpT:400.0 s
```

```
(x,y,v)=( 670, 501,   0) (RA,Dec)=(12:30:47.04,12:23:11.7) (R,G,B)=(  8,  0,  0)
```

```
                 / WFPC-II DATA DESCRIPTOR KEYWORDS
INSTRUME= 'WFPC2              ' / instrument in use
ROOTNAME= 'U2900103T          ' / rootname of the observ
FILETYPE= 'SCI                ' / shp, ext, edq, sdq, sc

                 / SCIENCE INSTRUMENT CONFIGURATION
MODE    = 'FULL               ' / instr. mode: FULL (ful
SERIALS = 'OFF                ' / serial clocks: ON, OFF

                 / IMAGE TYPE CHARACTERISTICS
IMAGETYP= 'EXT                ' / DARK/BIAS/IFLAT/UFLAT/
CDBSFILE= 'NO                 ' / GENERIC/BIAS/DARK/FLAT
PKTFMT  =                    96 / packet format code
DATE    = '27/02/94           ' / date file written (dd/

                 / FILTER CONFIGURATION
```

```
Building display array ...
```

1.2.3 Marking and Retrieving Data with StarView

Datasets are marked for retrieval by first clicking on them, then using the "Mark" button at the top of StarView. There is also the "All" button, which will mark all the datasets retrieved in the search. Marked datasets will be displayed in the Retrieval window. Datasets still within their proprietary period will be displayed in yellow, and users other than the proposal PI and those authorized by the PI will not able to retrieve them. The release date of files still within their proprietary period will also be indicated on the search results form.

If satisfied with the marked datasets, choose "Submit" in the Retrieval window to retrieve them. You will then be queried for both the type of data files associated with the dataset(s) to be retrieved, and for the method of delivery of these files. The options for type of file include files calibrated with the On-The-Fly-Recalibration (OTFR) pipeline for the WFPC2, NICMOS, STIS and ACS instruments. OTFR applies the best available calibration files (i.e., dark current and flat field images taken closest in time to the observations) to the uncalibrated data files. You may also request the uncalibrated (raw) files and calibration files separately. For some of the earlier instruments, e.g., WFPC and FOS, you may request both the calibration files actually applied to the images, as well as those that should provide the best calibration of them, if recalibration is desired. You may also request Data Quality and Observation Log files.

Options for data delivery include ftp transfer by the user from the HDA staging disk, automatic transfer from the HDA via the Internet to a directory specified by the user, and the mailing of tapes or disks. If Internet delivery is specified, you will be queried for the name of the computer and directory in which the files are to be placed, as well as your user name and password on that computer (these requests are encrypted, so there is no danger of your login information being stolen). Upon final submission of the request, you will receive an e-mail message acknowledging its receipt, and another message after all the requested files have been transferred. The status of the request, i.e., how many files have been transferred and any errors that have occurred, can be checked on a Web page that will be given in the acknowledgment message.

1.2.4 Using StarView to Retrieve Calibration Files and Proposal Information

StarView allows several additional types of searches of the HDA besides the Quick Search option described above. These can be selected from the Searches menu bar at the top of the StarView screen. One such search option is by instrument. This is necessary for identifying calibration reference files. As an example, selecting the option "WFPC2 OTFR" under the Instrument and WFPC2 sub-menus of the Searches menu, and then entering “M87” under Target Name in the qualifications box, brings up the screen shown in figure 1.3. This screen shows all the calibration images and files applied by OTFR to the first file in the set of WFPC2 images of M87, as well as whether the application of these files was performed or omitted in the calibration pipeline. This is the same set of images found by the Quick Search query described above, and the same information for the other datasets from this search can be found using the Previous, Next and Scan buttons. Once these calibration images have been identified, further information on them can be obtained. For example, taking the name of the flat field file found in the above search and entering it into the “WFPC2 Calibration Data” Searches option will retrieve information on when and where this file was taken, and the date after which its use is recommended. This will help users decide if they would prefer to recalibrate their data using different files.

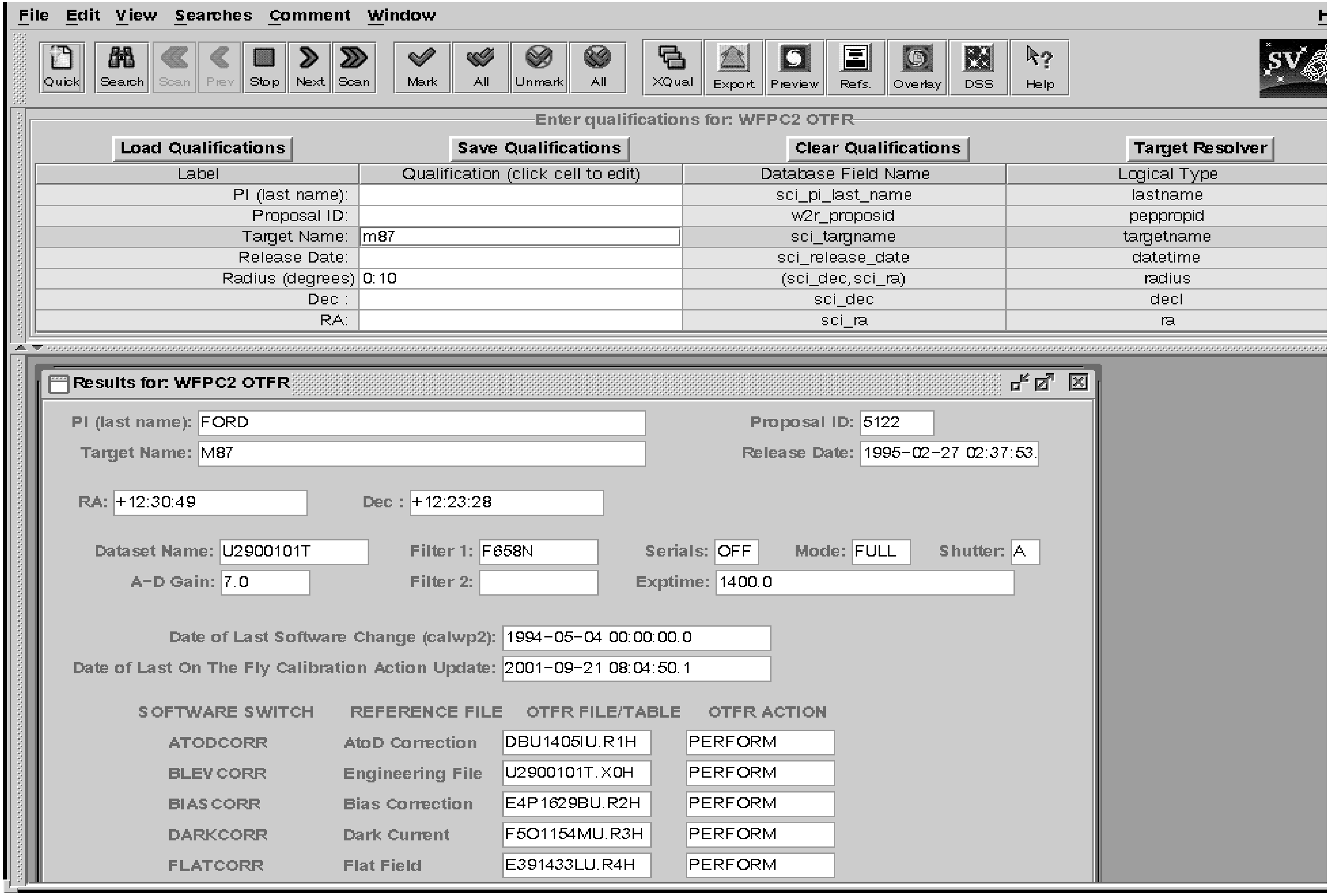
StarView can also be used to search for and view the abstracts of accepted HST proposals. Like the Preview capability of StarView, this provides additional information about a given dataset and whether it may be useful for your science goals. Viewing proposal abstracts is an option under the Searches menu, and an example is shown in figure 1.4. The Qualifications window again offers several parameters by which this search can be constrained, including proposal I.D. number, HST cycle, P.I. name, and combinations thereof. In the example shown only the proposal I.D. number was used.

Finally, StarView can be used during the Phase I proposal process to see whether or not HST observations of a given object or object class have already been made, or else are scheduled for execution. Specifically, the Duplications option under the Searches menu allows users to check a database containing both HDA files and a list of queued observations in order to see if a given object has been or will be observed. Similarly, under Duplications the user may also query the database of proposal abstracts for a given object or object class, to check for archived or scheduled observations.

1.2.5 Advanced Features of StarView

In addition to its basic search and retrieval function, StarView allows users to cross-qualify results from separate searches of the HDA, and to export the results of searches to disk as ASCII files. These operations are performed with the “XQual” and “Export” buttons, respectively. As an example of cross-qualification, a user might want to identify all the spiral galaxies for which both WFPC2 images and STIS spectra have been obtained. This could be accomplished with the Cross-Qualification feature by first doing two separate Quick Searches, in which these respective instruments are specified in the query box, and in which “Galaxy;Spiral” is typed in the Target Description box for both searches. Clicking the XQual button, specifying “Target Name” as the common field in the two sets of search results (as shown in figure 1.5), and clicking the “X-Qualify” button then identifies the galaxies occurring in both lists. StarView then places these galaxy names in the Target Name box of a new Quick Search window. Clicking the Search button with “WFPC2,STIS” entered for Instrument then gives a list of all the WFPC2 and STIS datasets for these galaxies. The Cross-Qualification function can also be performed on the files produced by the Export feature.

Figure 1.3: Results of StarView search for WFPC2 OTFR calibration files for M87



Enter qualifications for: WFPC2 OTFR

Label	Qualification (click cell to edit)	Database Field Name	Logical Type
PI (last name):		sci_pi_last_name	lastname
Proposal ID:		w2r_proposid	pepproposid
Target Name:	m87	sci_targname	targetname
Release Date:		sci_release_date	datetime
Radius (degrees)	0:10	(sci_dec,sci_ra)	radius
Dec :		sci_dec	decl
RA:		sci_ra	ra

Results for: WFPC2 OTFR

PI (last name): FORD
Proposal ID: 5122
Target Name: M87
Release Date: 1995-02-27 02:37:53.

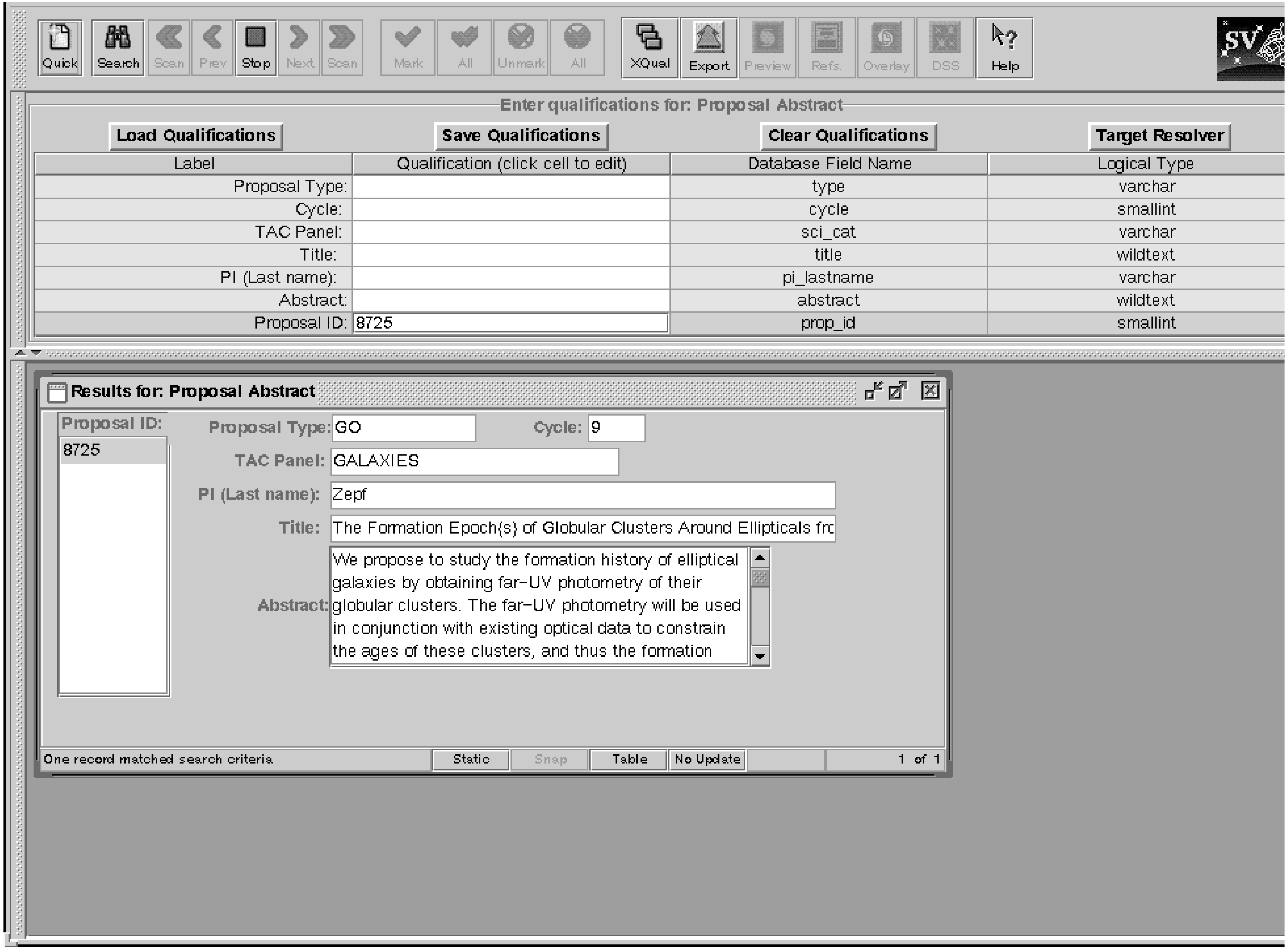
RA: +12:30:49
Dec : +12:23:28

Dataset Name: U2900101T
Filter 1: F658N
Serials: OFF
Mode: FULL
Shutter: A
A-D Gain: 7.0
Filter 2:
Exptime: 1400.0

Date of Last Software Change (calwp2): 1994-05-04 00:00:00.0
Date of Last On The Fly Calibration Action Update: 2001-09-21 08:04:50.1

SOFTWARE SWITCH	REFERENCE FILE	OTFR FILE/TABLE	OTFR ACTION
ATODCORR	AtoD Correction	DBU1405IU.R1H	PERFORM
BLEVCORR	Engineering File	U2900101T.X0H	PERFORM
BIASCORR	Bias Correction	E4P1629BU.R2H	PERFORM
DARKCORR	Dark Current	F5O1154MU.R3H	PERFORM
FLATCORR	Flat Field	E391433LU.R4H	PERFORM

Figure 1.4: Results of the StarView search for the abstract of Proposal 8725



Enter qualifications for: Proposal Abstract

Label	Qualification (click cell to edit)	Database Field Name	Logical Type
Proposal Type:		type	varchar
Cycle:		cycle	smallint
TAC Panel:		sci_cat	varchar
Title:		title	wildtext
PI (Last name):		pi_lastname	varchar
Abstract:		abstract	wildtext
Proposal ID:	8725	prop_id	smallint

Results for: Proposal Abstract

Proposal ID: 8725

Proposal Type: GO Cycle: 9

TAC Panel: GALAXIES

PI (Last name): Zepf

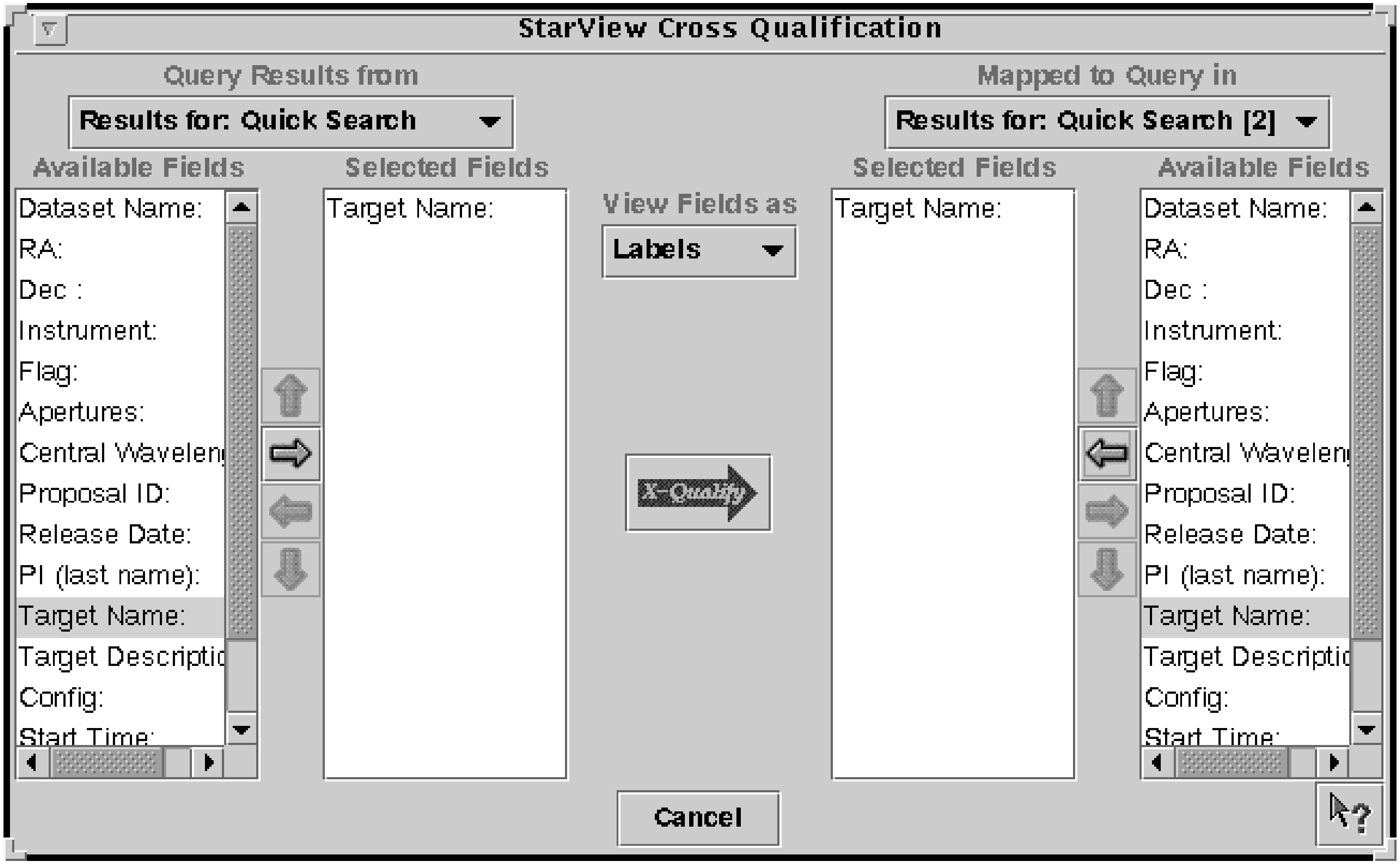
Title: The Formation Epoch(s) of Globular Clusters Around Ellipticals frc

Abstract: We propose to study the formation history of elliptical galaxies by obtaining far-UV photometry of their globular clusters. The far-UV photometry will be used in conjunction with existing optical data to constrain the ages of these clusters, and thus the formation

One record matched search criteria

1 of 1

Figure 1.5: Example of Cross Correlation Feature, in which Target Name has been chosen as the common feature to search for in two Quick Search result lists.



StarView Cross Qualification

Query Results from: Results for: Quick Search

Mapped to Query in: Results for: Quick Search [2]

View Fields as: Labels

Selected Fields: Target Name:

Selected Fields: Target Name:

1.2.6 StarView and the Visual Target Tuner

The Visual Target Tuner (VTT) is part of the Astronomer's Proposal Tool (APT) package, which has been created to aid astronomers in planning their HST observations during the Phase I and Phase II proposal stages (see http://apt.stsci.edu). VTT is an image display tool which allows the user to display DSS images or local FITS images with proper World Coordinate System keywords in the headers. It offers more features than JIPA, which is the default StarView display tool. However, for a limited number of operating systems, the VTT can be used with StarView. VTT offers the particular advantage that it can overlay the instrument apertures of multiple observations on a single DSS image. Clicking on these apertures will also highlight the associated datasets in StarView.

Currently, to combine StarView and VTT requires downloading and installing APT from the above Web site. APT is only available for those operating systems with the Java Virtual Machine 1.3 (JVM 1.3). You can download the StarView/VTT package with JVM 1.3 included (a large download), or if you already have JVM 1.3 installed, you can get the smaller APT/VTT package. To make StarView use VTT, you must change your Viewer options from JIPA to VTT. Go to the Environment sub-menu of Edit in StarView, and change JIPA to VTT in the Viewers section. If VTT is not listed here, you should reinstall the two programs. Following this change, the Preview, DSS and Overlay buttons of StarView should all bring up VTT.

Once VTT has been installed, you can also bring up StarView from it. Enter in to StarView mode by clicking on the StarView button in the lower left hand menu of VTT. Clicking on a DSS image will then spawn a Quick StarView screen with the R.A. and Dec of the position you clicked loaded into the search fields. You can enter other constraints into these fields as usual. Search results can be displayed on the VTT screen by selecting the results in StarView, and pressing the Overlay button.

1.2.7 Quick Data Retrieval with StarView

The following steps summarize the basic process that PIs need to go through to retrieve their data with StarView. These steps follow registration as a MAST user, notification from STScI that the observations for a given proposal are complete, and providing StarView with your e-mail information. They are intended as a quick reference for this process.

1. Start StarView.
2. Click the “Quick” button.
3. Enter your PI name and/or proposal ID number in the appropriate cell.
4. Click the “Search” button.

5. Use the “Scan” button to step through the retrieved files, after toggling the right most button at the bottom of the Results window to “Update,” to verify that all datasets have been retrieved.

6. Preview some or all of the datasets if desired, to verify data quality and target acquisition.

7. Click “All” to mark all datasets for retrieval, or “Mark” to mark individual datasets for retrieval.

8. Click “Submit” in the window that will be spawned by marking the files.

9. Enter your MAST username and password and specify the means of data delivery. StarView remembers your name and password from past searches so it does not have to be entered each time.

10. Click “Done”, and your data are on their way. You will receive an e-mail message when your retrieval has been queued, and another when the transfer is complete.

1.3 Getting Data With the World Wide Web

HDA datasets can be searched for, previewed and retrieved via the World Wide Web in very much the same way as with StarView. As noted in section 1.1, StarView offers more capabilities for this process, including cross-qualification, the use of VTT, and more information about instrument calibration files. However, Web retrievals may be preferable in some cases, particularly when information on calibration files is not needed, and the hypertext on the Results pages makes it easy to access all the information they contain. The starting point for Web-based searches of the HDA is the MAST web site at: http://archive.stsci.edu[3]

This web page is shown in figure 1.6. A powerful feature of MAST is that all of its mission archives, including the HDA, can be searched simultaneously. This is done with the Cross-Correlation Target Search option shown on the MAST home page. This search will return all datasets for all missions available for a given object or coordinates, according to the search constraints specified by the user (based on the wavelength region of interest), and will provide hypertext links to these datasets. If only HST datasets are desired, they can be accessed separately by clicking “HST” on the MAST home page.

[3] European archive users should generally use the ST-ECF Archive at http://archive.eso.org. Canadian users should request public archival data through the CADC web site at http://cadcwww.dao.nrc.ca. Proprietary data are only available through STScI.

The HST section of MAST offers tutorials about the HDA as well as a FAQ page and HDA news. It also provides links to HST Prepared datasets such as the Hubble Deep Field images. Clicking on the “Main Search Form” option of the HST section brings up the page shown in figure 1.7. Here the user is queried for the same search parameters as requested by StarView, e.g., Object Name, Instrument and Proposal I.D. Once these are entered, clicking the Search button returns a page listing the datasets found, which can then be selectively marked for retrieval. The data type and retrieval options remain the same as those for StarView. Previews of GIF files of the datasets are also available.

1.4 Reading HST Data Tapes and Disks

If you request HDA files on tapes or disks, you will receive them within a few weeks of your request. The tapes will contain **tar** files containing the requested datasets. The datasets will all be in FITS (Flexible Image Transport System) format[4]. You should thus first create a directory where you want your data to reside, e.g., /home/myname/myhstdata, go to that directory, then read the tape or disk using the Unix/Linux **tar** command to read the FITS files into it.

Currently, datasets obtained with HST's original instruments (FGS, FOC, FOS, GHRS, HSP and WFPC) as well as WFPC2 must have their FITS files converted to GEIS (Generic Edited Information Set) format in order to work on them with IRAF/STSDAS. Further information on HST file formats is presented in chapter 2. STSDAS is the package analysis software for HST data, and is discussed further in chapter 3. Datasets obtained with the other current HST instruments (ACS, NICMOS and STIS) should be reduced in FITS format without conversion to GEIS. STSDAS support for the analysis of WFPC2 data in FITS format is currently planned.

The steps for reading and converting FITS files to GEIS files are as follows:

First bring up IRAF/STSDAS in your IRAF home directory by typing

```
> cl
```

[4] A description of FITS format and various supporting documents can be found at the Web site http://fits.gsfc.nasa.gov/fits_home.html

This will start an IRAF session. IRAF and STSDAS are organized into *packages*. To load a package, type its name. To begin with, you must load the **stsdas** and **fitsio** (FITS Input/Output) packages:

```
cl> stsdas
st> fitsio
```

The IRAF prompt (such as st>) shows the first two letters of the most recently loaded package. The **fitsio** package contains the STSDAS programs (called *tasks* in the IRAF/STSDAS environment) required to read and write FITS files to and from tapes and disks. The two principle tasks are **strfits** for reading files, and **stwfits** for writing them.

Next, set the IRAF environment variable *imtype* to specify that your data files are to be written in GEIS format. This is done by typing

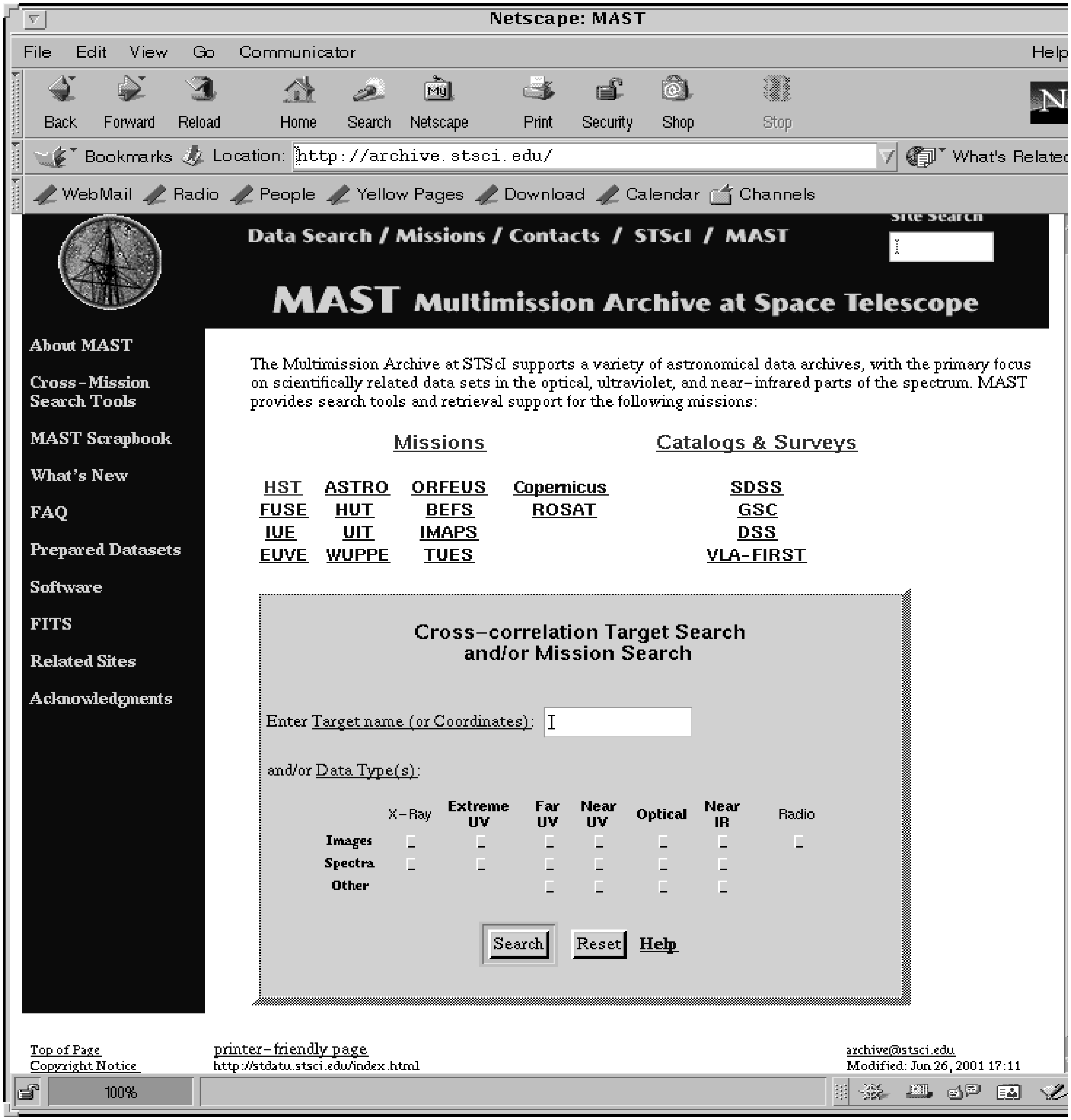
```
fi> set imtype="hhh"
```

You should then move to the directory containing the FITS files.

The last step is to use **strfits** to read the data. Like most IRAF/STSDAS tasks, **strfits** has several parameters that control its function. You can either edit these tasks using the IRAF "epar" command, or specify them on the command line. For the purpose converting FITS files to GEIS files, the important parameter is *oldirafname*, which needs to be set to "yes" in order to keep the file rootname the same. To convert all the FITS files in a directory to GEIS files, type

```
fi> strfits *.fits "  " oldirafname=yes
```

Figure 1.6: MAST Home Page


Netscape: MAST
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://archive.stsci.edu/ What's Relatec
WebMail Radio People Yellow Pages Download Calendar Channels
Data Search / Missions / Contacts / STScI / MAST
Site Search
MAST Multimission Archive at Space Telescope
About MAST
Cross-Mission Search Tools
MAST Scrapbook
What's New
FAQ
Prepared Datasets
Software
FITS
Related Sites
Acknowledgments
Search
Reset
Help
Top of Page
Copyright Notice
printer-friendly page
100%

The Multimission Archive at STScI supports a variety of astronomical data archives, with the primary focus on scientifically related data sets in the optical, ultraviolet, and near-infrared parts of the spectrum. MAST provides search tools and retrieval support for the following missions:

Missions				Catalogs & Surveys
HST	ASTRO	ORFEUS	Copernicus	SDSS
FUSE	HUT	BEFS	ROSAT	GSC
IUE	UIT	IMAPS		DSS
EUVE	WUPPE	TUES		VLA-FIRST

Cross-correlation Target Search and/or Mission Search

Enter Target name (or Coordinates):

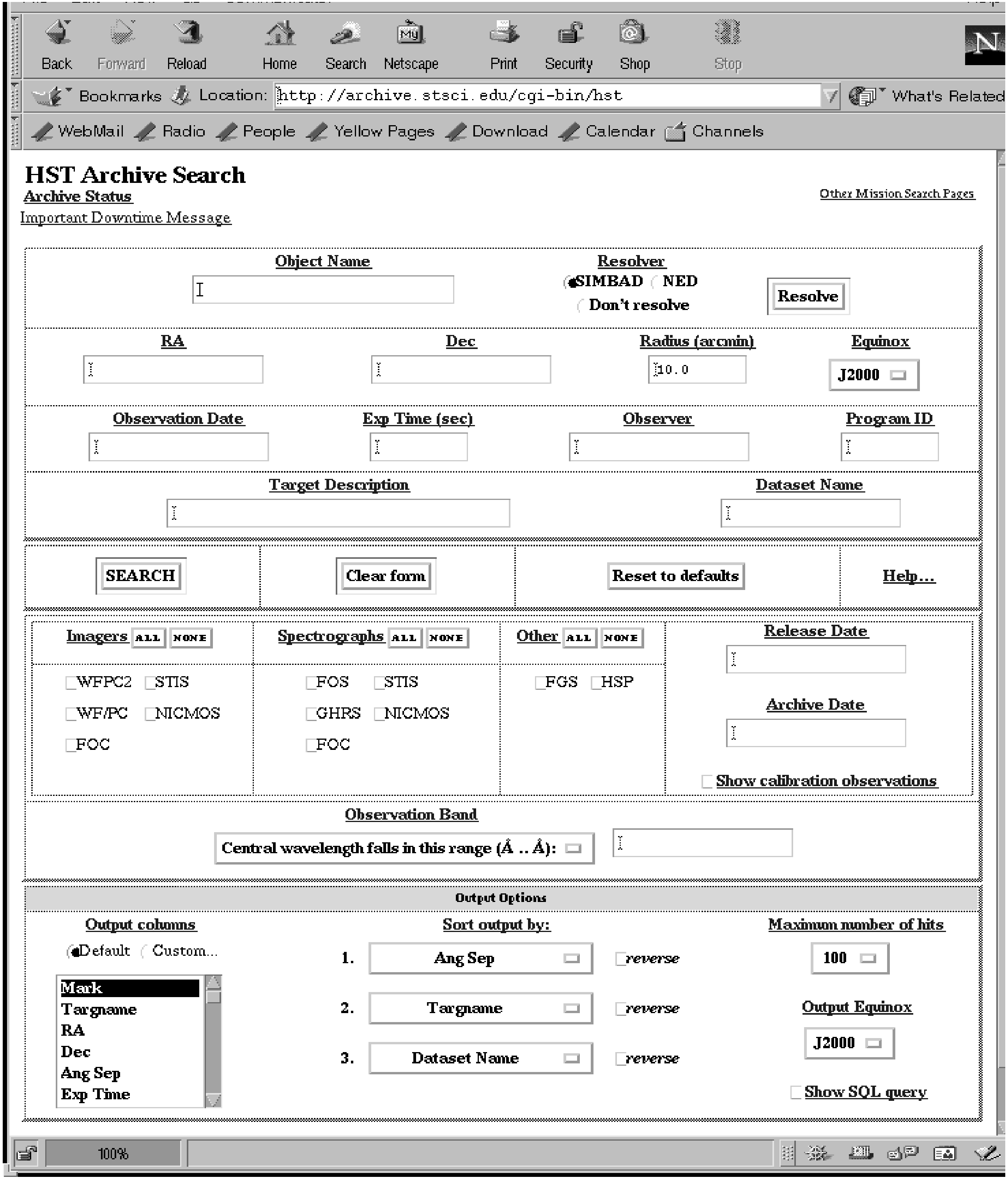
and/or Data Type(s):

	X-Ray	Extreme UV	Far UV	Near UV	Optical	Near IR	Radio
Images	□	□	□	□	□	□	□
Spectra	□	□	□	□	□	□	
Other			□	□	□	□	

http://stdatu.stsci.edu/index.html

archive@stsci.edu
Modified: Jun 26, 2001 17:11

Figure 1.7: HST Archive Web search Form



Location: http://archive.stsci.edu/cgi-bin/hst

HST Archive Search

Archive Status

Important Downtime Message

Other Mission Search Pages

Object Name:

Resolver: ● SIMBAD ○ NED ○ Don't resolve [Resolve]

RA: | Dec: | Radius (arcmin): 10.0 | Equinox: J2000

Observation Date: | Exp Time (sec): | Observer: | Program ID:

Target Description: | Dataset Name:

[SEARCH] [Clear form] [Reset to defaults] Help...

Imagers ALL NONE: □ WFPC2 □ STIS □ WF/PC □ NICMOS □ FOC

Spectrographs ALL NONE: □ FOS □ STIS □ GHRS □ NICMOS □ FOC

Other ALL NONE: □ FGS □ HSP

Release Date:

Archive Date:

□ Show calibration observations

Observation Band: Central wavelength falls in this range (Å .. Å):

Output Options

Output columns: ● Default ○ Custom...

- Mark
- Targname
- RA
- Dec
- Ang Sep
- Exp Time

Sort output by:

1. Ang Sep □ reverse
2. Targname □ reverse
3. Dataset Name □ reverse

Maximum number of hits: 100

Output Equinox: J2000

□ Show SQL query

This command will make GEIS format copies (having extension ".hhh") of all the FITS files in the directory, with the same rootname. Following reduction and analysis of the GEIS files with the IRAF/STSDAS tasks, they may be written back into FITS format, on hard disk or to a tape or other storage media, with the **stwfits** task.

CHAPTER 2:

HST File Formats

In this chapter. . .

STScI automatically processes and calibrates all the data received from HST. The suite of software programs that performs this processing—part of a system known as OPUS—is frequently called the *pipeline*, and its purpose is to provide data to observers and to the HST Data Archive in a form suitable for most scientific analyses. Pipeline processing assembles data received from HST into *datasets*, calibrates the data according to standard procedures described in the instrument sections of this handbook, and stores both calibrated and uncalibrated datasets in the Archive.

Pipelines of older instruments (FOC, FOS, FGS, GHRS, HSP, WF/PC-1, and WFPC2) generate files in the so-called GEIS (stands for Generic Edited Information Set) format. Since GEIS is a machine dependent format, these files are converted to a specific kind of FITS file format, sometimes referred as “waiver” FITS, before being archived. We'll explain the structure of this “waiver” FITS format later in this chapter. Since the “waiver” FITS format is only designed for archival purpose, it is necessary to convert it back to the GEIS format before further data processing and analysis using IRAF/STSDAS tasks.

Instruments installed after the 1997 servicing mission (STIS, NICMOS, ACS, and most likely all future instruments) have pipelines which generate FITS files directly. They are ready to be used by relevant IRAF/STSDAS tasks and, unlike the “waiver” FITS files, do NOT need to (and indeed, should not) be converted to GEIS format. Sometimes FITS files for the newer instruments are referred to as “FITS with extension” or “extended”

FITS files. But this can be misleading, since a “waiver” FITS file also has one (ASCII table) extension.

Much confusion has occurred about the two kinds of FITS files been archived at STScI. So we like to repeat one more time:

Older instruments (FOC, FOS, FGS, GHRS, HSP, WF/PC-1, and WFPC2) generate files in GEIS formats, but are stored and delivered as “waiver” FITS format in the archive, and need to be converted back to GEIS format before processing. Newer instruments (STIS, NICMOS, ACS) generate and store files in FITS format and should not be converted to GEIS.

This chapter describes these two HST file formats, first giving some historical perspective on the reasons why they were selected, then explaining the FITS and GEIS formats in more detail. STIS,ACS, and NICMOS observers should pay particular attention to the section on FITS files, which shows how to identify and access the contents of these files and covers some important conventions regarding header keywords. Veteran observers with the other instruments will find little new in the section on GEIS files, but newcomers to the older HST instruments should consult the material on data groups and conversion from FITS to GEIS before proceeding to chapter 3 of the HST Introduction.

2.1 Historical Perspective

In the early 1980’s, when GEIS was selected as the standard format for HST data files, it held several advantages over both FITS and the original IRAF format (OIF):

- GEIS allows floating-point data. The early incarnations of FITS accommodated only integer data, and this restriction to integers would have made data reduction and storage of calibrated data rather cumbersome.
- GEIS files can hold multiple images, each with associated parameters. This feature allowed the packaging of images from the four WF/PC-1 chips into a single unit, as well as the packaging of multiple FOS or GHRS readouts into single files. OIF files and early FITS files could contain only single images.

- GEIS data are stored in two parts, an ASCII header and a binary data file. The separation of these two pieces and the restriction of the header to ASCII made these headers easier to read and print in the days when computers were less powerful and tasks for reading header information were less numerous. OIF headers combine ASCII and binary information, and FITS headers come packaged with the data in a single file.

GEIS was also the standard format for archiving and distribution of HST data until September 1994, when the Space Telescope Data Archive and Distribution Service (ST-DADS) came online. This new system stores and distributes HST data files in machine-independent FITS format, but observers with FOC, FOS, FGS, GHRS, HSP, WF/PC-1, and WFPC2 still must convert their files to machine-dependent GEIS format as described in section 2.3.1 before using IRAF/STSDAS software (see chapter 3 in the HST Introduction) to reduce their data.

Since the selection of GEIS as HST's standard data format, FITS has added features that have dramatically increased its flexibility. In particular, FITS files can now contain multiple *image extensions*, each with its own header, size, and datatype, that allow multiple exposures to be packaged into the same file, along with associated error and data quality information. The FITS image kernel in IRAF version 2.11, released in August 1997, enables users to access FITS image extensions in ways similar to how they would access GEIS data groups.

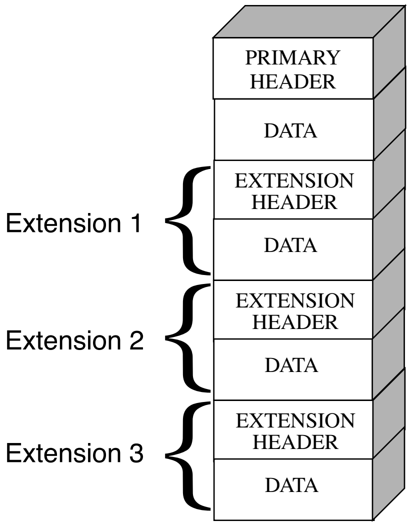
Because of these advantages, FITS was chosen as the standard reduction and analysis format for STIS and NICMOS. The STSDAS tasks written for these instruments expect FITS files as input and produce FITS files as output. *You cannot convert STIS and NICMOS files to GEIS format*. Observers using these instruments should therefore read the following section, which explains how to work with these new FITS files.

2.2 FITS File Format

Flexible Image Transport System (FITS) is a standard format for exchanging astronomical data between institutions, independent of the hardware platform and software environment. A data file in FITS format consists of a series of Header Data Units (HDUs), each containing two components: an ASCII text header and the binary data. The header contains a series of *header keywords* that describe the data in a particular HDU and the data component immediately follows the header.

The first header in a FITS file is known as the *primary header*, and any number of *extensions* can follow the primary HDU. The data unit following the primary header must contain either an image or no data at all, but each extension can contain one of several different data types, including images, binary tables, and ASCII text tables. The value of the XTENSION keyword in the extension's header identifies the type of data the extension contains. Figure 2.1 schematically illustrates the structure of a FITS file and its extensions.

Figure 2.1: FITS File Structure



The three-letter identifier (e.g., d0h) that follows the rootname of an HST data file (see appendix B for more on HST file names) has often been called an "extension" in the past. However, because of the potential for confusion with FITS extensions, this handbook will refer to these three-letter identifiers as "suffixes."

2.2.1 Working with FITS Image Extensions

The FITS image kernel included in IRAF version 2.11 is designed to read and write the images in FITS extensions and their associated headers. Once IRAF has ingested a FITS image and its header, it treats the header-data pair like any other IRAF image. The following discussion describes how to specify the image extensions in FITS files that you would like to process with IRAF/STSDAS tasks and presumes that you are using IRAF 2.11 or higher. It covers how to:

- List a FITS file's extensions.
- Access data in particular FITS extensions.

- Inherit keywords from the primary header.
- Append new extensions to existing FITS files.

Retaining the .fits at the end of every FITS file name in your file specifications will ensure that IRAF both reads and writes these images in FITS format.

If you want to work with STIS and NICMOS data, you will need to upgrade to IRAF 2.11 or higher and STSDAS 2.0.

Generating a FITS File Listing

Once you have downloaded STIS,ACS, or NICMOS FITS files from the Archive, you may want an inventory of their contents. To generate a listing of a FITS file's extensions, you can use the **catfits** task in the **tables** package. The following example, in table 2.1, illustrates the first 11 lines generated by **catfits** from a NICMOS MULTIACCUM FITS file containing images only.

The first column of a **catfits** listing gives the extension numbers. Note that the primary HDU is labeled extension number zero. The second column lists the extension type, given by the keyword XTENSION (IMAGE = image, BINTABLE = binary table, TABLE = ASCII table). The third column lists the extension name, given by the keyword EXTNAME. In STIS, ACS, and NICMOS image files, the EXTNAME values SCI, ERR, and DQ indicate science, error, and data quality images, respectively. NICMOS image files contain samples and exposure time images as well, with EXTNAME values SAMP and TIME.

Each STIS or NICMOS readout generates an image set or *imset*. STIS and ACS imsets comprise three images (SCI, ERR, DQ), while NICMOS imsets comprise five (SCI, ERR, DQ, SAMP, TIME). All images belonging to the same imset share the same integer value of the EXTVER keyword, given in the fourth column of a **catfits** listing. Several STSDAS tasks can work with entire imsets (see section 3.3.3), but most operate on individual images. See the Data Structure chapters of STIS, ACS, and NICMOS Data Handbooks for more information on the contents of imsets.

Table 2.1: NICMOS MULTIACCUM Listing from catfits

```
tt> catfits n3t501c2r_raw.fits

EXT#  FITSNAME      FILENAME              EXTVE  DIMENS     BITPI OBJECT

0     n3t501c2r_raw n3t501c2r_raw.fits                      16    n3t501c2r_raw.f
1       IMAGE       SCI                   1     256x256     16    n3t501c2r_raw.f
2       IMAGE       ERR                   1                 -32
3       IMAGE       DQ                    1                 16
4       IMAGE       SAMP                  1                 16
5       IMAGE       TIME                  1                 -32
6       IMAGE       SCI                   2     256x256     16
7       IMAGE       ERR                   2                 -32
8       IMAGE       DQ                    2                 16
9       IMAGE       SAMP                  2                 16
10      IMAGE       TIME                  2                 -32
```

Accessing FITS Images

After you have identified which FITS image extension you wish to process, you can direct an IRAF/STSDAS task to access that extension using the following syntax:

fitsfile.fits[*extension number*][*keyword options*][*image section*]

Note that all the bracketed information is optional. However, the only time it is *valid* to provide only a file name without further specification is when the file is a simple FITS file that contains a single image in the primary HDU.

Designation of the extension number is the most basic method of access, but it is not necessarily the most helpful. Referring to an extension's EXTNAME and EXTVER in the `[keyword options]` is often more convenient. If a number follows an EXTNAME, IRAF interprets the number as an EXTVER. For example, if extension number 6 holds the science image belonging to the imset with EXTVER = 2, as in the **catfits** listing on the previous page, you can specify it in two equivalent ways:

```
fitsfile.fits[6]
fitsfile.fits[sci,2]
```

Designations giving an EXTNAME without an EXTVER refer to the first extension in the file with the specified value of EXTNAME. Thus, `fitsfile.fits[sci]` is the same as `fitsfile.fits[sci,1]`.

The syntax for designating image sections adheres to the IRAF standard, so in the current example the specifications

```
fitsfile.fits[6][100:199,100:299]
fitsfile.fits[sci,2][100:199,100:299]
```

both extract a 100 by 200 pixel subsection of the same science image in

```
fitsfile.fits.
```

Header Keywords and Inheritance

STIS, ACS, and NICMOS data files use an IRAF image kernel convention regarding the relationship of the primary header keywords to image extensions in the same file. In particular, IRAF allows image extensions to *inherit* keywords from the primary header under certain circumstances. When this inheritance takes place, the primary header keywords are practically indistinguishable from the extension header keywords. This feature circumvents the large scale duplication of keywords that share the same value for all extensions. The primary header keywords effectively become global keywords for all image extensions. The FITS standard does not cover or imply keyword inheritance, and while the idea itself is simple, its consequences are often complex and sometimes surprising to users.

Generally keyword inheritance is the default, and IRAF/STSDAS applications will join the primary and extension headers and treat them as one. For example, using **imheader** as follows on a FITS file will print both primary and extension header keywords to the screen:

```
cl> imheader fitsfile.fits[sci,2] long+ | page
```

Using **imcopy** on such an extension will combine the primary and extension headers in the output HDU, even if the output is going to an extension of another FITS file. Once IRAF has performed the act of inheriting the primary header keywords, it will normally turn the inheritance feature off in any output file it creates unless specifically told to do otherwise.

If you need to change the value of one of the global keywords inherited from the primary header, you must edit the primary header itself (i.e., “extension” [0]).

Keyword inheritance is not always desirable. For example, if you use **imcopy** to copy all the extensions of a FITS file to a separate output file, IRAF will write primary header keywords redundantly into each extension header. You can suppress keyword inheritance by using the NOINHERIT keyword in the file specification. For example:

```
im> imcopy fitsfile.fits[6][noinherit] outfile.fits
im> imcopy fitsfile.fits[sci,2,noinherit] outfile.fits
```

Both of the preceding commands will create an output file whose header contains only those keywords that were present in the original extension header. Note that in the second command, the `noinherit` specification is bracketed with the EXTNAME and EXTVER keywords and not in a separate bracket of its own, as in the first command where an absolute extension number is used. For a complete explanation of FITS file name specifications, see:

http://iraf.noao.edu/iraf/web/docs/fitsuserguide.html.

Appending Image Extensions to FITS Files

IRAF/STSDAS tasks that produce FITS images as output can either create new FITS files or append new image extensions to existing FITS files. You may find the following examples useful if you plan to write scripts to reduce STIS, ACS, or NICMOS data:

If the specified output file does not yet exist, a new output file is created containing only a primary HDU if no specification is appended to the output file name. For example, to copy the contents of the primary header of `fitsfile.fits` into the primary HDU of the FITS file `outfile.fits`, type the command:

```
cl> imcopy fitsfile.fits[0] outfile.fits
```

If the specified output file already exists and you want to append a new extension to it, you need to include the APPEND option in the output file specification. The following command appends extension `[sci,2]` of `fitsfile.fits` onto the existing file `outfile.fits`, while retaining the original EXTNAME and EXTVER of the extension—the `noinherit` specification inhibits the copying of the primary header keywords from the input file into the output extension header:

```
    cl> imcopy fitsfile.fits[sci,2,noinherit] \
    >>> outfile.fits[append]
```

If you want to change the EXTNAME or EXTVER of the appended extension, you can specify the new values of these keywords in the output extension, like this:

```
cl> imcopy fitsfile.fits[sci,2,noinherit] \
>>> outfile.fits[sci,3,append]
```

For obvious reasons, it is not generally advisable for two file extensions in the same FITS file to share the same EXTNAME and EXTVER values. However, if you must append an extension to an output file already containing an extension with the same EXTNAME/EXTVER pair you can do so with the DUPNAME option:

```
cl> imcopy fitsfile.fits[7] \
>>> outfile.fits[append,dupname]
```

If you need to replace an existing extension with a new output extension, you can use the OVERWRITE option as follows. Overwriting can cause a lengthy rewrite of the whole file to insert the new extension, if its size is not the same as the extension it replaces.

```
cl> imcopy fitsfile.fits[sci,2,noinherit] \
>>> outfile.fits[sci,2,overwrite]
```

2.2.2 Working with FITS Table Extensions

STIS and NICMOS use FITS tables in two basic ways. Both instruments produce association tables (see appendix B.3) listing the exposures that go into constructing a given association product. In addition, STIS provides certain spectra, calibration reference files, and time-tagged data in tabular form . Here we describe:

- How to access and read FITS table extensions.
- How to specify data arrays in FITS table cells.

This discussion assumes you are using STSDAS 2.0 or later. (The IRAF FITS kernel deals only with FITS images. The **tables** package installed with STSDAS handles FITS table extensions.)

Accessing FITS Tables

You can access data in FITS table extensions using the same tasks appropriate for any other STSDAS table, and the syntax for accessing a specific FITS table is similar to the syntax for accessing FITS images (see section 2.2.1), with the following exceptions:

- The FITS table interface does not support header keyword inheritance.
- FITS tables cannot reside in the primary HDU of a FITS file. They must reside instead in a FITS table extension, in either ASCII form (XTENSION=TABLE) or binary form (XTENSION=BINTABLE).
- If the first extension in a FITS file is a TABLE or a BINTABLE, you can access it by typing the file name with no extension specified. It is not sufficient for the table to be just the first BINTABLE or TABLE; *it must actually be the first extension*.

For example, running **catfits** on the NICMOS association table `n3tc01010_asn.fits` provides the following output:

```
fi> catfits n3tc01010_asn.fits


EXT#  FITSNAME       FILENAME              EXTVE ...

0     n3tc01010_asn  N3TC01010_ASN.FITS ...

1      BINTABLE    ASN                  1 ...
```

Extension number 1 holds the association table, which has EXTNAME=ASN and EXTVER=1. You can use the **tprint** task in the STSDAS **tables** package to print the contents of this table, and the following commands are all equivalent:

```
tt> tprint n3tc01010_asn.fits
tt> tprint n3tc01010_asn.fits[1]
tt> tprint n3tc01010_asn.fits[asn,1]
```

STSDAS **tables** tasks can read both FITS TABLE and BINTABLE extensions, but they can write tabular results only as BINTABLE extensions. Tasks that write to a table in-place (i.e., **tedit**) can modify an existing FITS extension, and tasks that create a new table (i.e., **tcopy**) will create a new extension when writing to an existing FITS file. If the designated output file does not already exist, the task will create a new FITS file with the output table in the first extension. If the output file already exists, your task will append the new table to the end of the existing file; the APPEND option necessary for appending FITS image extensions is not required. As with FITS images, you can specify the EXTNAME and EXTVER of the output extension explicitly, if you want to assign them values different from those in the input HDU. You can also specify the

OVERWRITE option if you want the output table to supplant an existing FITS extension. For example, you could type:

```
tt> tcopy n3tc01010_asn.fits out.fits[3][asn,2,overwrite]
```

This command would copy the table in the first extension of `n3tc01010_asn.fits` into the third extension of `out.fits`, while reassigning it the EXTNAME/EXTVER pair `[asn,2]` and overwriting the previous contents of the extension. Note that overwriting is the only time when it is valid to specify an extension, EXTNAME, and an EXTVER in the output specification.

Specifying Arrays in FITS Table Cells

A standard FITS table consists of columns and rows forming a two-dimensional grid of cells; however, each of these cells can contain a data array, effectively creating a table of higher dimensionality. Tables containing extracted STIS spectra take advantage of this feature. Each column of a STIS spectral table holds data values corresponding to a particular physical attribute, such as wavelength, net flux, or background flux. Each row contains data corresponding to one spectral order, and tables holding echelle spectra can contain many rows. Each cell of such a spectral table can contain a one-dimensional data array corresponding to the physical attribute and spectral order of the cell.

In order to analyze tabular spectral data with STSDAS tasks other than the **sgraph** task and the **igi** package, which have been appropriately modified, you will need to extract the desired arrays from the three-dimensional table. Two new IRAF tasks, named **tximage** and **txtable**, can be used to extract the table-cell arrays. Complementary tasks, named **tiimage** and **titable**, will insert arrays back into table cells. To specify the arrays which should be extracted from or inserted into the table cells, you will need to use the *selectors* syntax to specify the desired row and column. The general syntax for selecting a particular cell is:

```
intable.fits[extension number][c:column_selector][r:row_selector]

or

intable.fits[keyword options][c:column_selector][r:row_selector]
```

A *column selector* is a list of column patterns separated by commas. The column pattern is either a column name, a file name containing a list of column names, or a pattern using the IRAF pattern matching syntax (type `help system.match`, for a description of the IRAF pattern matching syntax). If you need a list of the column names, you can run the **tlcol** task (type `tlcol infile.fits`).

Rows are selected according to a *filter*. The filter is evaluated at each table row, and the row is selected if the filter is true. For example, if you specify:

```
infile.fits[3][c:WAVELENGTH,FLUX][r:SPORDER=(68:70)]
```

IRAF will extract data from the table stored in the third extension of the FITS file, `infile.fits`, specifically the data from the columns labelled WAVELENGTH and FLUX, and will restrict the extraction to the rows where the spectral order (SPORDER) is within the range 68–70, inclusive. Alternatively, if you specify:

```
infile.fits[sci,2][c:FLUX][r:row=(20:30)]
```

IRAF will obtain data from the table stored in the FITS file extension with an EXTNAME of SCI and EXTVER of 2. The data will come from the column FLUX and be restricted to the row numbers 20–30, inclusive. Eventually, all STSDAS and TABLES tasks will be able to use row and column selection. For a complete explanation of the table selector syntax, `type help selectors`.

2.3 GEIS File Format

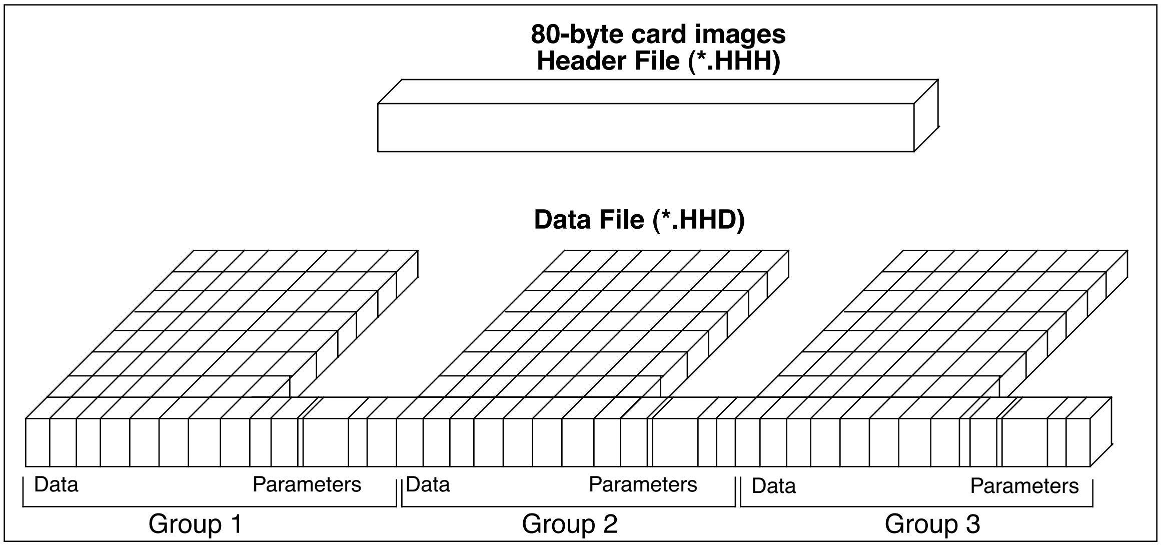
The HST-specific Generic Edited Information Set (GEIS) format[1] is the standard format for reducing data from FOC, FOS, FGS, GHRS, HSP, WF/PC-1, and WFPC2. All HST images in GEIS format consist of two components: a *header file* and a separate *binary data file*, both of which should reside in the same directory. GEIS header files, whose suffixes end in "h" (e.g., `w0lo0105t.c1h`), consist entirely of ASCII text in fixed-length records of 80 bytes. These records contain header keywords that specify the properties of the image itself and the parameters used in executing the observation and processing the data. GEIS binary data files, whose suffixes end in "d" (e.g., `w0lo0105t.c1d`), contain one or more *groups* of binary data. Each group comprises a data array followed by an associated block of binary parameters called the Group Parameter Block (GPB). The sizes and datatypes of the data arrays and group parameters in

[1] GEIS files are also commonly referred to as STSDAS images.

each group of a GEIS file are identical. Figure 2.2 depicts the structure of a GEIS data file graphically.

The binary content of GEIS files is machine dependent. Copying GEIS files directly from one platform to another (e.g., from a VAX to a Sun) may result in unreadable data.

Figure 2.2: GEIS File Structure



2.3.1 Converting FITS to GEIS

The STScI archive stores and distributes datasets from FOC, FOS, FGS, GHRS, HSP, WF/PC-1, and WFPC2 in a special archival FITS format. *We highly recommend that users convert these datasets back into their native GEIS format before working with them*. Your data must be in GEIS format for you to use many of the STSDAS software tools developed specifically for analysis of these data. It is important to use the **strfits** task found in **stsdas.fitsio** or in **tables.fitsio** to perform the conversion from archival FITS format to the GEIS format because the data-processing pipeline employs a special convention for mapping GEIS files to FITS format. While other FITS readers may be able to read portions of the data correctly, they are unlikely to reconstruct the entire data file properly.

To recreate the original multigroup GEIS file using strfits, you must first type:

```
cl> set imtype=hhh
```

This command tells IRAF to write output files in GEIS format. You then need to set the **strfits** parameters `xdimtogf` and `oldirafname` both to "yes". For example, after you have set `imtype` = `hhh`, you can convert the FITS file `*_hhf.fits` into the GEIS format files `*.hhh` and `*.hhd` by typing:

```
cl> strfits *_hhf.fits "" xdim=yes oldiraf=yes
```

2.3.2 GEIS Data Groups

One of the original advantages of GEIS format noted in Section 2.1 was that it could accommodate multiple images within a single file. This feature is useful because a single HST observation often produces multiple images or spectra. For example, a single WF/PC-1 or WFPC2 exposure generates four simultaneous images, one for each CCD chip. Likewise, the FOS and GHRS obtain data in a time-resolved fashion so that a single FOS or GHRS dataset comprises many spectra—one corresponding to each readout. The data corresponding to each sub-image (for the WF/PC-1 or WFPC2) or each sub-integration (for the FOS or GHRS) are stored sequentially in the groups of a single GEIS binary data file. The header file corresponding to this data file contains the information that applies to the observation as a whole (i.e., to all the groups in the image), and the group-specific keyword information is stored in the group parameter block of each data group in the binary data file.

The *number* of groups produced by a given observation depends upon the instrument configuration, the observing mode, and the observing parameters. Table 2.2 lists the *contents* and the number of groups in the final calibrated image for the most commonly-used modes of each instrument which uses the GEIS data format.

Table 2.2: Groups in Calibrated Images, by Instrument and Mode

Instrument	Mode	Number of Groups	Description
FGS	All	7	FGS data are not reduced with IRAF and STSDAS. Therefore, FGS groups have different meaning than for the other instruments.
FOC	All	1	All FOC images have only a single group.
FOS	ACCUM	*n*	Group *n* contains accumulated counts from groups (subintegrations) 1, 2, ... *n*. The last group is the full exposure.
	RAPID	*n*	Each group is an independent subintegration with exposure time given by group parameter EXPOSURE.

Instrument	Mode	Number of Groups	Description
HSP	All	1	HSP datasets always have only a single group that represents either digital star (`.d0h`, `.c0h`), digital sky (`.d1h`, `.c1h`), analog star (`.d2h`, `.c2h`), or analog sky (`.d3h`, `.c3h`).
GHRS	ACCUM	*n*	Each group is an independent subintegration with exposure time given by group parameter EXPOSURE. If FP-SPLIT mode was used, the groups will be shifted in wavelength space. The independent subintegrations should be coadded prior to analysis.
	RAPID	*n*	Each group is a separate subintegration with exposure time given by group parameter EXPOSURE.
WF/PC-1	WF	4	Group *n* represents CCD chip *n*, e.g., group 1 is chip 1 (unless not all chips were used). Group parameter DETECTOR always gives chip used.
	PC	4	Group *n* is chip *n* + 4, e.g., group 1 is chip 5. If not all chips were used, see the DETECTOR parameter which always gives the chip used.
WFPC2	All	4	Planetary chip is group 1, detector 1. Wide Field chips are groups 2–4 for detectors 2–4. If not all chips were used, see the DETECTOR keyword.

2.3.3 Working with GEIS Files

This section briefly explains how to work with information in GEIS header and data files.

GEIS Headers

Header keyword information relevant to each group of a GEIS file resides in two places, the header file itself and the parameter block associated with the group. Because GEIS header files are composed solely of ASCII text, they are easy to print using standard Unix or VMS text-handling facilities. However, the group parameters are stored in the binary data file. To access them you need to use a task such as **imheader**, as shown in section“Printing Header Information”.

You can use the IRAF **hedit** task to edit the keywords in GEIS headers. While it is possible to edit GEIS header files using standard Unix and VMS text editors, you must maintain their standard 80-character line length. The **hedit** task automatically preserves this line length. If you need to add or delete group parameters, you can use the STSDAS **groupmod** task in the **stsdas.hst_calib.ctools** package. The STSDAS **chcalpar** task, described in more detail in the Calibration chapters for each instrument’s data handbook, is useful for updating header keywords containing calibration switches and calibration reference files.

Always edit headers using tasks like hedit, eheader, and chcalpar. Editing headers with a standard text editor may corrupt the files by creating incorrect line lengths.

GEIS Data Files

Numerous IRAF/STSDAS tasks exist for working with GEIS images (see chapter 3 of the HST Introduction). Most of these tasks operate on only one image at a time, so you usually need to specify which group of a GEIS file is to be processed. If you do not specify a group, your task will choose the first group by default.

Specifying a Group

To specify a particular group in a GEIS file, append the desired group number in square brackets to the file name (e.g., `z2bd010ft.d0h[10]`). For example, to apply the **imarith** task to group 10 of a GEIS image, type the following (always refer to a GEIS file by its header file name, i.e. *.??h, even though mathematically you are operating on the data portion):

```
cl> imarith indata.hhh[10] + 77.0 outdata.hhh
```

This command will add 77.0 to the data in group 10 of the file `indata.hhh`, and will write the output to a new single-group file called `outdata.hhh`. Any operation performed on a single group of a multigroup GEIS file results in an output file containing a single group.

Specifying an Image Section

If you wish to process only a portion of an image, you can specify the image section after the group specification in the following manner:

```
cl> imarith indata.hhh[2][100:199,200:399] * 32.0 outdata.hhh
```

This command extracts a 100 by 200 pixel subsection of the image in the second group of the file `indata.hhh`, multiplies this data by a factor of 32.0, and stores the result in a new output file, `outdata.hhh`, which is a 100 by 200 pixel single group GEIS file.

Printing Header Information

As discussed in the previous section, the task **imheader** extracts and prints information about the GEIS image. This task reports the image name, dimensions (including the number of groups), pixel type, and title of the image when it is run in default mode. For example:

```
cl> imhead indata.hhh
    indata.hhh[1/64][500][real]: INDATA[1/64]
```

The output line indicates that `indata.hhh` is a multigroup GEIS file which contains 64 groups of images, each consisting of a spectral array 500 pixels in length. The data type of the values is real (floating point). Note that since no group designation was provided, the task defaulted to the first group. To reveal more information regarding group 10, you can type:

```
cl> imhead indata.hhh[10] long+ | page
```

which will generate a long listing of both the ASCII header parameters in the `*.hhh` file and the specific group parameters for group 10 from the `*.hhd` file.

Other Group-Related Tasks

Currently, IRAF or STSDAS tasks cannot process all the groups in an input image and write the results to corresponding groups in an output image. However, there are several STSDAS tasks, particularly in the **toolbox.imgtools** and **hst_calib.ctools** packages, that simplify working with group format data. Please refer to chapter 3 and the *STSDAS User's Guide* for more details about working with GEIS images.

2.3.4 The "waiver" FITS format

Although "waiver" is not quite the accurate or good word for the intended purpose, for historic reasons it has stuck and will be reluctantly adopted. However, in the past, a grammatically incorrect word "waivered" had been used.

The "waiver" FITS format was developed when the HST archive needed a format to store and distribute the data products in a machine-independent medium for the community, at a time before FITS image extension was standardized. As a result, the "waiver" FITS format was adopted as a compromise.

Since, at the time, FITS could only have a single image while the HST data (in GEIS format) may have several images as multiple groups in one file, the idea is to stack the images of different groups together as a new dimension in the FITS image. As for group parameters, they are put in an ASCII table and the table becomes the first (and only) extension of the FITS file.

For example, the WFPC2 pipeline generates the science data as a GEIS file of 4 groups, each is an 800x800 image corresponding to one of the 4 detectors. When this GEIS file is converted to the "waiver" FITS file, the FITS file has an image of 800x800x4 (a three-dimensional image!) at its primary HDU. Similarly, an FOS GEIS file may have 40 groups, each group is a 1-D image (spectrum) of the size 2064. The waiver FITS file then will have one 2-D image of the size 2064x40, at its primary HDU. In the case of WFPC2, the first extension of the waiver FITS file will be an ASCII table containing 4 rows; each row corresponds to a group. The value of each group parameter is under a column named after the group parameter, i. e. the value of the group parameter CRVAL1 of the 2nd group will be at the 2nd row, under the column named "CRVAL1". In other words, the ASCII table has as many rows as there are groups in the original GEIS file, and as many columns as group parameters.

Although, *in theory, certain* IRAF/STSDAS tasks can directly access the data in the "waiver" FITS file, e.g. to display the 2nd "group" of a WFPC2 image:

```
st.> display u67m0206r_c0f.fits[0][*,*,2]
```

will work, while *most tasks, especially those specific to HST instruments, can not*. It is therefore HIGHLY recommended that all waiver FITS files are converted back to the GEIS format, by using the task **strfits**, before further processing and analysis with IRAF/STSDAS tasks.

CHAPTER 3:

STSDAS Basics

In this chapter. . .

The Space Telescope Science Data Analysis System (STSDA) is the software system for calibrating and analyzing data from the Hubble Space Telescope. The package contains programs—called *tasks*—that perform a wide range of functions supporting the entire data analysis process, from reading tapes, through reduction and analysis, to producing final plots and images. This chapter introduces the basics of STSDAS, showing you how to display your data, leading you through some simple data manipulations, and pointing you towards more sophisticated tasks, some of which are described in the instrument data handbooks.

STSDAS is layered on top of the Image Reduction and Analysis Facility *(IRAF)* software developed at the National Optical Astronomy Observatory (NOAO). Any task in IRAF can be used in STSDAS, and the software is portable across a number of platforms and operating systems. To exploit the power of STSDAS effectively, you need to know the basics of IRAF. If you are not already familiar with IRAF, consult the IRAF Primer in Appendix A before reading further.

3.1 Navigating STSDAS

The tasks in STSDAS are far too numerous and complicated to describe comprehensively in this volume. Instead, we will show you where to find the STSDAS tasks appropriate for handling certain jobs. You can refer to online help or the *STSDAS User's Guide* for details on how to use these tasks. Some useful online help commands are:

- `help` *task* - provides detailed descriptions and examples of each task.
- `help` *package* - lists the tasks in a given package and their functions.
- `describe` *task* - provides a detailed description of each task.
- `example` *task* - provides examples of each task.
- `apropos` *word* - searches the online help database for tasks relating to the specified word (see figure A.4).

3.1.1 STSDAS Structure

STSDAS is structured so that related tasks are grouped together as packages. For example, tasks used in the calibration process can be found in the **hst_calib** package and tasks used for image display and plotting can be found in the **graphics** pack. Figure 3.1 shows the STSDAS package structure. Note that IRAF version 2.11 must be installed on your system in order for you to use STSDAS 2.0 and TABLES version 2.0 or higher

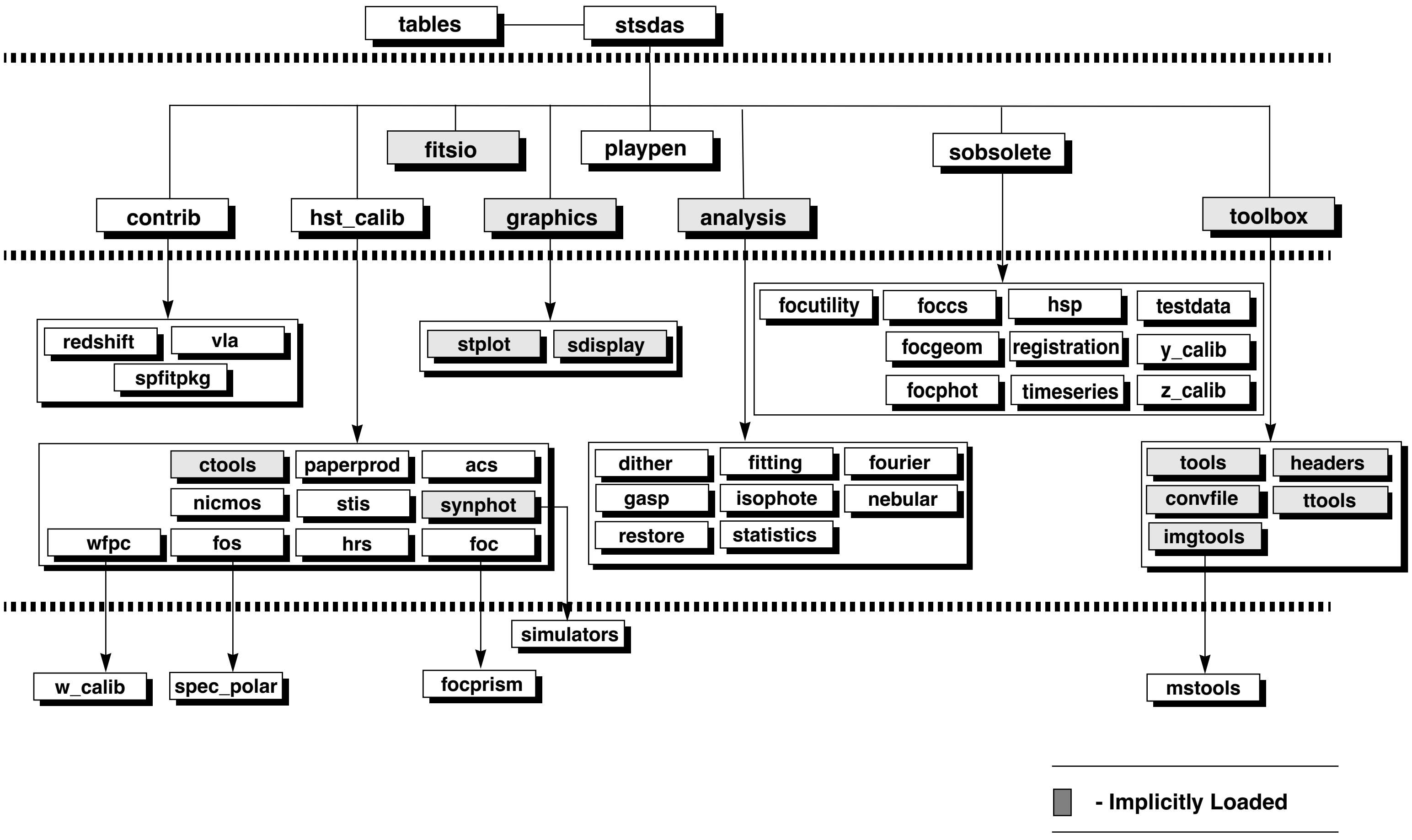
3.1.2 Packages of General Interest

Images

Both IRAF and STSDAS contain a large number of tasks that work with HST images. Some of the packages you should investigate are:

- **images**: This package includes general tasks for copying (**imcopy**), moving (**imrename**), and deleting (**imdelete**) image files. These tasks operate on both the header and data portions of the image. The package also contains a number of general purpose tasks for operations such as rotating and magnifying images.
- **stsdas.toolbox.imgtools**: This package contains general tools for working with multigroup GEIS images, including tasks for working with masks, and general purpose tasks for working with the pixel data, such as an interactive pixel editor (**pixedit**).

Figure 3.1: STSDAS Version 2.3 Package Structure



- **stsdas.toolbox.imgtools.mstools**: This package contains tools for working with FITS image extensions, in particular STIS and NICMOS image sets (imsets).
- **stsdas.analysis**: This package contains general tasks for image analysis, such as Fourier analysis and dither.

Tables

Several of the analysis packages in STSDAS, including calibration pipeline tasks, create output files in STSDAS table format, which is a binary row-column format, or in FITS binary table format. (ASCII-format tables are also supported, for input only.) The *STSDAS User's Guide* describes the STSDAS table format in detail. Tasks in the **ttools** package or in the external **tables** package can be used to read, edit, create, and manipulate tables. For example:

- **tread** displays a table, allowing you to move through it with the arrow keys.
- **tprint** displays a table.
- **tcopy** copies tables.
- **tedit** allows you to edit a table.

Many other tasks in **ttools** perform a variety of other functions. See the online help for details.

3.2 Displaying HST Images

This section will be of interest primarily to observers whose datasets contain two-dimensional images, as it explains:

- How to display images in IRAF using the **display** task.
- How to display subsections of images.

Observers viewing WF/PC-1 and WFPC2 data may wish to remove cosmic rays before displaying their data. The FOC photon-counting hardware does not detect cosmic rays at easily as CCDs, the NICMOS pipeline automatically removes cosmic rays from MULTIACCUM observations, and the STIS pipeline automatically removes cosmic rays from CR-SPLIT association products.

3.2.1 The display Task

The most general IRAF task for displaying image data is the **display** task, the best choice for a first look at HST imaging data. To display an image, you need to:

1. Start an image display server, such as SAOimage, in a separate window from your IRAF session, either from a different xterm window or as a background job before starting IRAF. To start SAOimage, type the following:

```
saoimage &
```

2. Load the **images.tv** package from the window where you're running IRAF:

```
cl> images
im> tv
```

Several different display servers, including SAOimage, ds9 (the next generation of SAOimage), and Ximtool, can be used with IRAF. ds9 may be retrieved from http://hea-www.harvard.edu/RD/ds9/. Ximtool may be retrieved from ftp://iraf.noao.edu/iraf/x11iraf/.

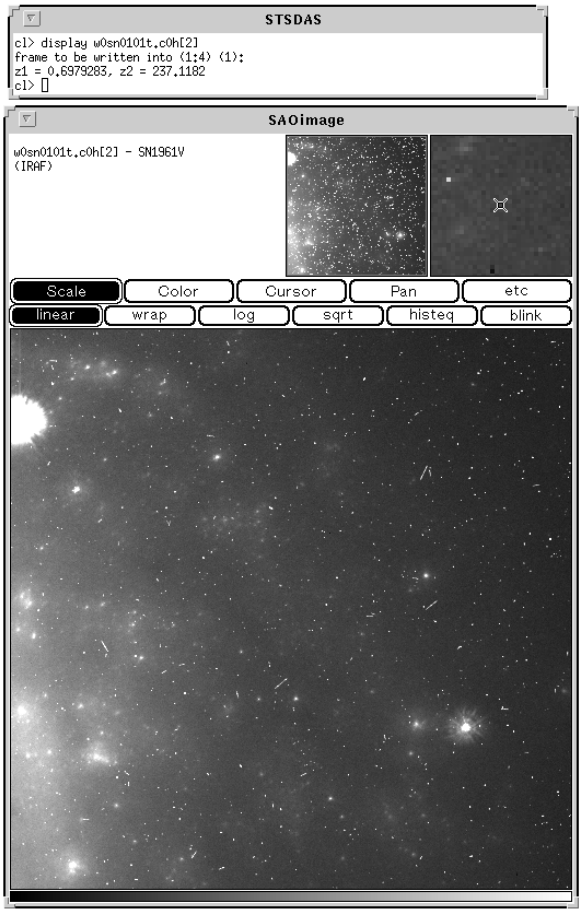
3. Display the image with the IRAF **display** task, using the syntax appropriate for the file format (Chapter 2 explains how to specify GEIS groups and FITS extensions):

```
tv> display fname.c0h[2] 1 (GEIS group 2)
tv> display fname.fits[11] 1 (FITS extension 11)
tv> display fname.fits[sci,3] 1 (FITS extension sci,3)
```

Note that when using **display** or any other task on GEIS images, you do not need to specify a group; the first group is the default. However, when working with FITS files you must specify an extension, unless the FITS file contains only a single image in the primary data unit and has no extensions. figure 3.2 shows how to display group two of a WF/PC-1 image.

If you want to display all four chips of a WF/PC-1 or WFPC2 image simultaneously, you can create a mosaic with the STSDAS wmosaic task in the hst_calib.wfpc package. Type help wmosaic for details.

Figure 3.2: Displaying an Image



STSDAS

```
cl> display w0sn0101t.c0h[2]
frame to be written into (1:4) (1):
z1 = 0.6979283, z2 = 237.1182
cl> ▯
```

SAOimage

```
w0sn0101t.c0h[2] - SN1961V
(IRAF)
```

Run display task from IRAF window

z1 and z2 are image intensity range

Image appears in SAOimage window

To print hardcopy:

1. Click “etc”
2. Click “print”

Modifying the Display

There are two ways to adjust how your image is displayed:

- Use the SAOimage command buttons that control zooming, panning, etc.
- Reset the **display** task parameters.

Once an image appears in your SAOimage window, you can use the SAOimage commands displayed near the top of the image window to manipulate or print your image. The *SAOimage Users Guide* describes these commands, although most are fairly intuitive. Just click on the buttons to scale, pan, or print the image, or to perform other commonly-used functions. On-line help is also available at the system level: type `man saoimage` in Unix or `help saoimage` in VMS.

The example in figure 3.2 shows how you should display an image for a first look. By default, **display** automatically scales the image intensity using a sampling of pixels throughout the image. During your first look, you may want to experiment with the scaling using the `zscale`, `zrange`, `z1` and `z2` parameters. The `zscale` parameter toggles the autoscaling. Setting `zscale-` and `zrange+` tells the task to use minimum and maximum values from the image as the minimum and maximum intensity values. To customize your minimum and maximum intensity display values, set `zscale-`, `zrange-`, `z1` to the minimum value and `z2` to the maximum value that you want displayed. For example:

```
im> disp w0mw0507v.c0h 1 zrange- zscale- z1=2.78 z2=15.27
```

Notice in figure 3.2 that when you run **display**, the task shows you the `z1` and `z2` values that it calculates. You can use these starting points in estimating reasonable values for the minimum and maximum intensity display parameters.[1]

If you want to display an image with greater dynamic range, you may prefer to use logarithmic scaling. However, the log scaling function in SAOimage divides the selected intensity range into 200 linearly spaced levels before taking the log. The resulting intensity levels are rendered in a linear rather than logarithmic sense. You can often obtain better results if you create a separate logarithmic image to display. One way to create a logarithmic image is with the **imcalc** task:

```
im> imcalc x2ce0502t.c1h x2ce0502t.hhh “log10(im1+1.0)”
```

If the peak pixel in your original image contained 2000 counts, for example, you would then display the logarithmic image with `z1=0` and `z2=3.3`.

Otherwise, the user can simply do:

```
im> display  x2ce0502t.c1h ztrans=log
```

[1] Type `help display` within IRAF to obtain more information about these parameters.

The image display buffer can also be adjusted in IRAF by setting the stdimage parameter. For example,

```
im> set stdimage = imt 2048
```

will allow a larger image to be displayed without losing the borders.

3.2.2 Working with Image Sections

Sometimes you may want to display only a portion of an image, using the syntax for specifying image sections discussed in chapter 2. Your specified pixel range should give the starting point and ending point, with a colon separating the two. List the horizontal (*x* axis) range first, followed by the vertical (*y* axis) range. For example, to specify a pixel range from 101 to 200 in the *x* direction and all pixels in the *y* direction from group three of a GEIS format image:

```
tv> display image.hhh[3][101:200,*] 1
```

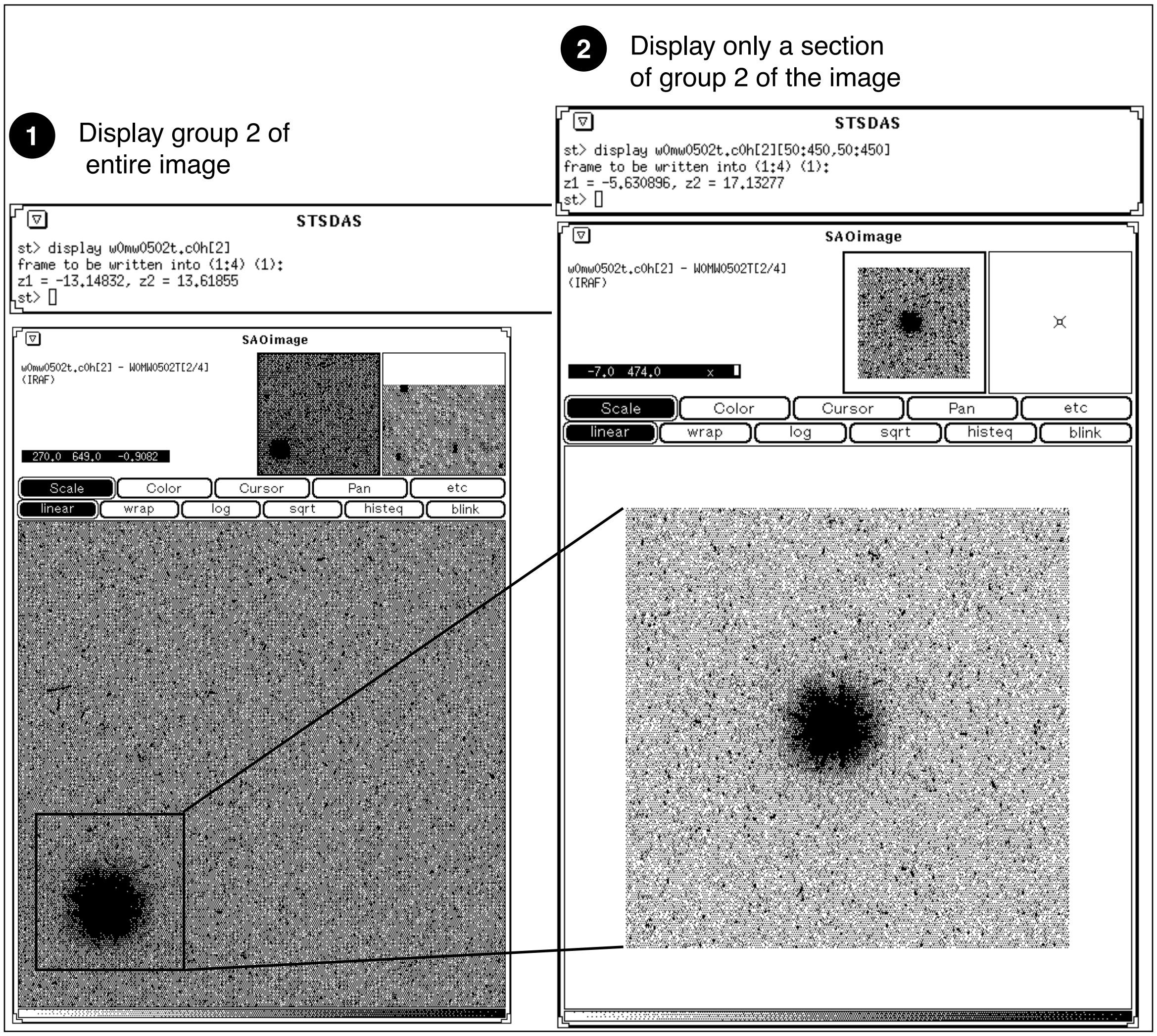
To specify the same pixel range in the second SCI extension of a NICMOS FITS image:

```
tv> display image.fits[sci,2][101:200,*] 1
```

If you specify both a group and an image section of a GEIS file, the group number must come first. When displaying sections of FITS image extensions, you must specify the extension, which also comes before the image section

Figure 3.3 shows examples of displaying an image and an image section.

Figure 3.3: Displaying Sections and Groups of an Image



1 Display group 2 of entire image

STSDAS

```
st> display w0mw0502t.c0h[2]
frame to be written into (1:4) (1):
z1 = -13.14832, z2 = 13.61855
st> []
```

SAOimage

```
w0mw0502t.c0h[2] - W0MW0502T[2/4]
(IRAF)
270.0  649.0  -0.9082
```

2 Display only a section of group 2 of the image

STSDAS

```
st> display w0mw0502t.c0h[2][50:450,50:450]
frame to be written into (1:4) (1):
z1 = -5.630896, z2 = 17.13277
st> []
```

SAOimage

```
w0mw0502t.c0h[2] - W0MW0502T[2/4]
(IRAF)
-7.0  474.0      x
```

3.3 Analyzing HST Images

This section describes methods for using STSDAS and IRAF to work with two-dimensional image data from HST. Subjects include:

- Relating your image to sky coordinates.
- Examining and manipulating your image.
- Working with STIS, ACS, and NICMOS imsets.
- Converting counts to fluxes.

3.3.1 Basic Astrometry

This section describes how to determine the orientation of an HST image and the RA and Dec of any pixel or source within it, including:

- Tasks that supply positional information about HST images.
- Methods for improving your absolute astrometric accuracy.

Positional Information

The header of every calibrated HST two-dimensional image contains a linear astrometric plate solution, written in terms of the standard FITS astrometry header keywords: CRPIX1, CRPIX2, CRVAL1, CRVAL2, and the CD matrix—CD1_1, CD1_2, CD2_1, and CD2_2. IRAF/STSDAS tasks can use this information to convert between pixel coordinates and RA and Dec. Two simple tasks that draw on these keywords to relate your image to sky coordinates are:

- **disconlab**: Displays your image with a superimposed RA and Dec grid. Simply open an SAOimage window and type, for example:

```
sd> disconlab n3tc01a5r_cal.fits[1]
```

- **xy2rd**: Translates *x* and *y* pixel coordinates to RA and Dec. (The task **rd2xy** inverts this operation.) SAOimage displays the current *x,y* pixel location of the cursor in the upper-left corner of the window. To find the RA and Dec of the current pixel, you supply these coordinates to **xy2rd** by typing

```
sd> xy2rd n3tc01a5r_cal.fits[1] x y
```

Table 3.1 lists some additional tasks that draw on the standard astrometry keywords.

Observers should be aware that these tasks do not correct for geometric distortion. Only FOC images currently undergo geometric correction during standard pipeline processing (the `.c0h/.c0d` and `.c1h/.c1d` FOC images have been geometrically corrected); STIS images will be geometrically corrected in the pipeline once suitable calibration files are in hand. If you need precise relative astrometry, you should use an instrument-specific task that accounts for image distortion, such as the **metric** task for WF/PC-1 and WFPC2 images.

Do not use tasks like rimcursor or xy2rd directly on WF/PC-1 or WFPC2 images if you require accurate relative positions. WF/PC-1 and WFPC2 pipelines do not correct for geometric distortions which will affect the accuracy of relative positions. Both wmosaic and metric, found in the stsdas.hst_calib.wfpc package, correct for this distortion.

Table 3.1: Additional IRAF and STSDAS Astrometry Tasks

Task	Purpose
compass	Plot north and east arrows on an image.
north	Display the orientation of an image based on keywords.
rimcursor	Determine RA and Dec of a pixel in an image.
wcscoords	Use WCS[1] to convert between IRAF coordinate systems.
wcslab	Produce sky projection grids for images.

[1] World Coordinate System (WCS). Type “help specwcs” at the IRAF prompt for details.

Improving Astrometric Accuracy

Differential astrometry (measuring a position of one object relative to another in an image) is easy and relatively accurate for HST images, while absolute astrometry is more difficult, owing to uncertainties in the locations of the instrument apertures relative to the Optical Telescope Assembly (OTA or V1) axis and the inherent uncertainty in Guide Star positions. However, if you can determine an accurate position for any single star in your HST image, then your absolute astrometric accuracy will be limited only by the accuracy with which you know that star’s location and the image orientation.

If there is a star on your image suitable for astrometry, you may wish to extract an image of the sky around this star from the Digitized Sky Survey and measure the position of that star using, for example, the GASP software (described in the *STSDAS User’s Guide*). These tools provide an absolute positional accuracy of approximately 0″.7. Contact the Help Desk for assistance (send E-mail to `help@stsci.edu`).

3.3.2 Examining and Manipulating Image Data

This section describes **implot** and **imexamine**, two basic IRAF tools for studying the characteristics of an image, and table 3.3 lists some useful IRAF/STSDAS tasks for manipulating images.

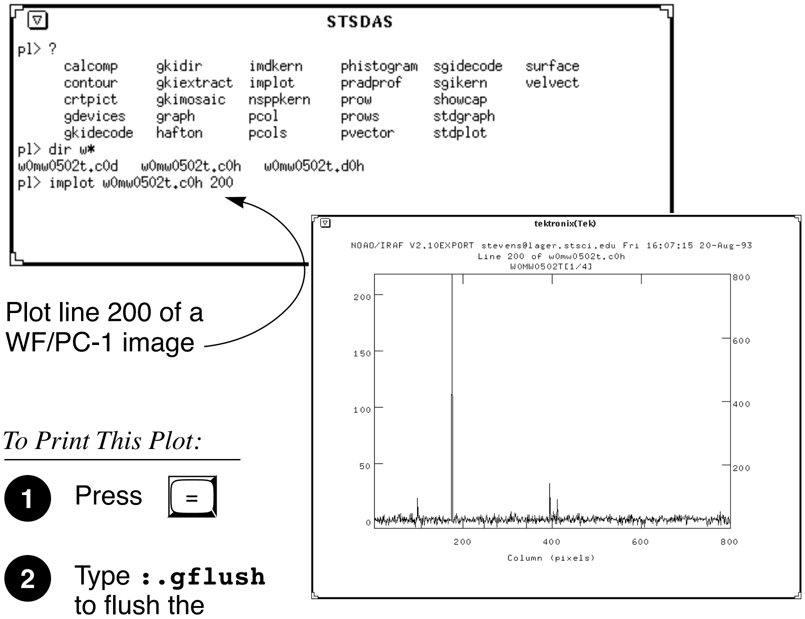
implot

The IRAF **implot** task (in the **plot** package) allows you to examine an image interactively by plotting data along a given *line* (*x* axis) or *column* (*y* axis). When you run the task, a large number of commands are available in addition to the usual cursor mode commands common to most IRAF plotting tasks. A complete listing of commands is found in the on-line help, but the most commonly used are listed in table 3.2. Figure 3.4 shows an example of how to use the **implot** task.

Table 3.2: Basic implot Commands

Keystroke	Command
?	Display on-line help.
L	Plot a line.
C	Plot a column.
Q	Quit implot.
J	Move down.
K	Move up.
Space	Display coordinates and pixel values.

Figure 3.4: Plotting Image Data with implot


STSDAS
tektronix(Tek)
NOAO/IRAF V2.10EXPORT stevens@lager.stsci.edu Fri 16:07:15 20-Aug-93
Line 200 of w0mw0502t.c0h
W0MW0502T[1/4]
200
150
100
50
0
800
600
400
200
200
400
600
800
Column (pixels)

```
pl> ?
      calcomp     gkidir      imdkern     phistogram  sgidecode   surface
      contour     gkiextract  implot      pradprof    sgikern     velvect
      crtpict     gkimosaic   nsppkern    prow        showcap
      gdevices    graph       pcol        prows       stdgraph
      gkidecode   hafton      pcols       pvector     stdplot
pl> dir w*
w0mw0502t.c0d   w0mw0502t.c0h   w0mw0502t.d0h
pl> implot w0mw0502t.c0h 200
```

Plot line 200 of a WF/PC-1 image

To Print This Plot:

1. Press =
2. Type **`:.gflush`** to flush the buffer

imexamine

The IRAF **imexamine** task (in the **images.tv** package) is a powerful task that integrates image display with various types of plotting capabilities. Commands can be passed to the task using the image display cursor and the graphics cursor. A complete description of the task and its usage are provided in the online help, available from within the IRAF environment by typing `help imexamine`.

Table 3.3: Image Manipulation Tasks

Task	Package	Purpose
boxcar	images.imfilter	Boxcar smooth a list of images
gcombine	stsdas.toolbox.imgtools	Combine images using various algorithms and rejection schemes
gcopy	stsdas.toolbox.imgtools	Copy GEIS multigroup images
geomap	images.immatch	Compute a coordinate transformation
geotran	images.immatch	Resample an image based on geomap output
grlist	stsdas.graphics.stplot	List of file names of all groups of a GEIS image (to make @lists)
gstatistics	stsdas.toolbox.imgtools	Compute image statistics[1]
imcalc	stsdas.toolbox.imgtools	Perform general arithmetic on GEIS images[a]
imedit	images.tv	Fill in regions of an image by interpolation
imexamine	images.tv	Examine images using display, plots, and text (see “imexamine” on page 3-13)
implot	plot	Plot lines and columns of images (see “implot” on page 3-12)
magnify	images.imgeom	Magnify an image
msarith	stsdas.toolbox.mstools	Performs basic arithmetic on STIS and NICMOS imsets
mscombine	stsdas.toolbox.mstools	Extension of **gcombine** for STIS and NICMOS imsets
msstatistics	stsdas.toolbox.mstools	Extension of **gstatistics** for STIS and NICMOS imsets
newcont	stsdas.graphics.stplot	Draw contours of two-dimensional data
pixcoord	stsdas.hst_calib.wfpc	Compute pixel coordinates of stars in a GEIS image
plcreate	xray.ximages	Create a pixel list from a region file (e.g., from SAOimage)
rotate	images.imgeom	Rotate an image
saodump	stsdas.graphics.sdisplay	Make image and colormap files from SAOimage display
siaper	stsdas.graphics.stplot	Plot science instrument apertures of HST

[1] Will process all groups of a multigroup GEIS file.

3.3.3 Working with STIS, ACS, and NICMOS Imsets

STIS, ACS, and NICMOS data files contain groups of images, called imsets, associated with each individual exposure. A STIS or ACS imset comprises SCI, ERR, and DQ images, which hold science, error, and data quality information. A NICMOS imset, in addition to its SCI, ERR, and DQ images, also contains TIME and SAMP images recording the integration time and number of samples corresponding to each pixel of the SCI image. See the STIS, ACS, and NICMOS Data Structures chapters for more details on imsets.

Here we describe several STSDAS tasks, located in the **stsdas.toolbox.imgtools.mstools** package, that have been designed to work with imsets as units and to deconstruct and rebuild them.

msarith

This tool is an extension of the IRAF task **imarith** to include error and data quality propagation. The **msarith** task supports the four basic arithmetic operations (+, -, *, /) and can operate on individual or multiple imsets. The input operands can be either files or numerical constants; the latter can appear with an associated error, which will propagate into the error array(s) of the output file. Table 3.4 below shows how this task operates on the SCI, ERR, and DQ images in a STIS, ACS, or NICMOS imset, as well as the additional TIME and SAMP images belonging to NICMOS imsets:

Table 3.4: Task msarith Operations

Operation	Operand2	SCI	ERR	DQ	TIME	SAMP
ADD	file	op1+op2	$\sqrt{\sigma 1^2 + \sigma 2^2}$	OR	T1+T2	S1+S2
SUB	file	op1-op2	$\sqrt{\sigma 1^2 + \sigma 2^2}$	OR	T1	S1
MULT	file	op1*op2	$(\mathrm{op1} \times \mathrm{op2})\sqrt{(\sigma 1/\mathrm{op1})^2 + (\sigma 2/\mathrm{op2})^2}$	OR	T1	S1
DIV	file	op1/op2	$(\mathrm{op1}/\mathrm{op2})\sqrt{(\sigma 1/\mathrm{op1})^2 + (\sigma 2/\mathrm{op2})^2}$	OR	T1	S1
ADD	constant	op1+op2	$\sqrt{\sigma 1^2 + \sigma 2^2}$	...	...	...
SUB	constant	op1-op2	$\sqrt{\sigma 1^2 + \sigma 2^2}$	...	...	...
MULT	constant	op1*op2	$(\mathrm{op1} \times \mathrm{op2})\sqrt{(\sigma 1/\mathrm{op1})^2 + (\sigma 2/\mathrm{op2})^2}$	...	T1*op2	...
DIV	constant	op1/op2	$(\mathrm{op1}/\mathrm{op2})\sqrt{(\sigma 1/\mathrm{op1})^2 + (\sigma 2/\mathrm{op2})^2}$	...	T1*op2	...

In table 3.4, the first operand (op1) is always a file, and the second operand (op2) can be either a constant or a file. The ERR arrays of the input files (σ1 and σ2) are added in quadrature. If the constant is given with an error (σ2), the latter is added in quadrature to the input ERR array. Note that in table 3.4 the pixels in the SCI images are in counts, but **msarith** can also operate on count rates.

mscombine

This task allows you to run the STSDAS task **gcombine** on STIS, ACS, and NICMOS data files. It divides each imset into its basic components (SCI, ERR, and DQ, plus SAMP and TIME for NICMOS) to make them digestible for **gcombine**. The SCI extensions become the inputs proper to the underlying **gcombine** task, and the ERR extensions become the error maps. The DQ extensions are first combined with a user-specified Boolean mask allowing selective pixel masking and then fed into the data quality maps. If scaling by exposure time is requested, the exposure times of each imset are read from the header keyword PIXVALUE in the TIME extensions.

Once **gcombine** has finished, **mscombine** then reassembles the individual output images into imsets and outputs them as one STIS, ACS, or NICMOS data file. The output images and error maps from **gcombine** form the SCI and ERR extensions of the output imset. The DQ extension will be a combination of the masking operations and the rejection algorithms executed in **gcombine**. For NICMOS, the TIME extension will be the sum of the TIME values from the input files minus the rejected values, divided on a pixel-by-pixel basis by the number of valid pixels in the output image. The final TIME array will be consistent with the output SCI image (average or median of the science data). The SAMP extension for NICMOS is built from all the input SAMP values, minus the values discarded by masking or rejection.

msstatistics

This tool is an extension of **gstatistics** in the STSDAS package, which is in turn an extension of **imstatistics**. The main novelty is the inclusion of the error and data quality information included with STIS, ACS, and NICMOS images in computing statistical quantities. In addition to the standard statistical quantities (min, max, sum, mean, standard deviation, median, mode, skewness, kurtosis), two additional quantities have been added to take advantage of the error information: the weighted mean and the weighted variance of the pixel distribution. If x_i is the value at the i-th pixel, with associated error σ_i, the weighted mean and variance used in the task are:

$$\langle x \rangle_w = \frac{\sum_i \frac{x_i}{\sigma_i \times \sigma_i}}{\sum_i \frac{1}{\sigma_i \times \sigma_i}}$$

and:

$$\langle \sigma \rangle_w^2 = \frac{1}{\sum_i \frac{1}{\sigma_i \times \sigma_i}}$$

The data quality information carried by the STIS, ACS, or NICMOS file is used to reject pixels in the statistical computation. Users can supply additional masks to reject objects or regions from the science arrays.

mssplit and msjoin

The **mssplit** task extracts user-specified imsets from a STIS, ACS, or NICMOS data file and copies them into separate files. Each output file contains a single imset along with the primary header of the original file. You might find this task useful for reducing the size of a STIS, ACS, or

NICMOS file containing many imsets or for performing analysis on a specific imset. The **msjoin** task inverts the operation of **mssplit**: it assembles separate imsets into a single data file.

There are additional tasks in this package for deleting and sorting imsets, as well as tasks for addressing a specific image class within an imset.

3.3.4 Photometry

Included in this section are:

- A list of IRAF/STSDAS tasks useful for determining source counts.
- Instructions on how to use header keyword information to convert HST counts to fluxes or magnitudes.
- A brief description of **synphot**, the STSDAS synthetic photometry package.

IRAF and STSDAS Photometry Tasks

The following are some useful IRAF/STSDAS packages and tasks for performing photometry on HST images:

- **apphot**: aperture photometry package.
- **daophot**: stellar photometry package useful for crowded fields.
- **isophote**: package for fitting elliptical isophotes.
- **imexamine**: performs simple photometry measurements.
- **imstat**: computes image pixel statistics.
- **imcnts**: sums counts over a specified region, subtracting background.
- **plcreate**: creates pixel masks.

Consult the online help for more details on these tasks and packages. The document “Photometry using IRAF” by Lisa A. Wells, provides a general guide to performing photometry with IRAF; it is available through the IRAF web page:

`http://iraf.noao.edu/docs/photom.html`

The apphot package allows you to measure fluxes within a series of concentric apertures. This technique can be used to determine the flux in the wings of the PSF, which is useful if you wish to estimate the flux of a saturated star by scaling the flux in the wings of the PSF to an unsaturated PSF.

Converting Counts to Flux or Magnitude

All calibrated HST images record signal in units of counts or Data Numbers (DN)[2]—NICMOS data is DN s^{-1}. The pipeline calibration tasks do not alter the units of the pixels in the image. Instead they calculate and write the inverse sensitivity conversion factor (PHOTFLAM) and the ST magnitude scale zero point (PHOTZPT) into header keywords in the calibrated data. WF/PC-1 and WFPC2 observers should note that the four chips are calibrated individually, so these photometry keywords belong to the group parameters for each chip.

For all instruments other than NICMOS, PHOTFLAM is defined to be the *mean* flux density F_λ in units of erg cm^{-2} s^{-1} $Å^{-1}$ that produces 1 count per second in the HST observing mode (PHOTMODE) used for the observation. If the F_λ spectrum of your source is significantly sloped across the bandpass or contains prominent features, such as strong emission lines, you may wish to recalculate the inverse sensitivity using **synphot**, described below. WF/PC-1 observers should note that the PHOTFLAM value calculated during pipeline processing does not include a correction for temporal variations in throughput owing to contamination buildup. Likewise, FOC observers should note that PHOTFLAM values determined by the pipeline before May 18, 1994 do not account for sensitivity differences in formats other than 512 x 512.

To convert from counts or DN to flux in units of erg cm^{-2} s^{-1} $Å^{-1}$, multiply the total number of counts by the value of the PHOTFLAM header keyword and divide by the value of the EXPTIME keyword (exposure time). You can use the STSDAS task **imcalc** to convert an entire image from counts to flux units. For example, to create a flux-calibrated output image `outimg.fits` from an input image `inimg.fits[1]` with header keywords PHOTFLAM = 2.5E-18 and EXPTIME = 1000.0, you could type:

```
st> imcalc inimg.fits[1] outimg.fits "im1*2.5E-18/1000.0”
```

Calibrated NICMOS data are in units of DN s^{-1}, so the PHOTFLAM values in their headers are in units of erg cm^{-2} $Å^{-1}$. You can simply multiply these images by the value of PHOTFLAM to obtain fluxes in units of erg cm^{-2} s^{-1} $Å^{-1}$. NICMOS headers also contain the keyword PHOTFNU in units of Jy s. Multiplying your image by the PHOTFNU value will therefore yield fluxes in Janskys.

[2] Except for 2-D rectified STIS images, which are in units of I_λ.

If your HST image contains a source whose flux you know from ground based measurements, you may choose to determine the final photometry of your HST image from the counts observed for this source.

To convert a measured flux *F*, in units of erg cm^{-2} s^{-1} $Å^{-1}$, to an ST magnitude, plug it into the following equation:

$m = -2.5 \times \log10\,(F) + \mathrm{PHOTZPT}$

where the value of the PHOTZPT keyword is the zero point of the ST magnitude scale. The zero point of the ST magnitude system has always been and probably always will be equal to –21.10, a value chosen so that Vega has an ST magnitude of zero for the Johnson *V* passband (see Koornneef et al., 1986; Horne, 1988; and the *Synphot Users Guide*).

synphot

The STSDAS synthetic photometry package, called **synphot**, can simulate HST observations of astronomical targets with known spectra. It contains throughput curves of all HST optical components, such as mirrors, filters, gratings, apertures, and detectors, and can generate passband shapes for any combination of these elements. It can also generate synthetic spectra of many different types, including stellar, blackbody, power-law and H II region spectra, and can convolve these spectra with the throughputs of HST's instruments. You can therefore use it to compare results in many different bands, to cross-calibrate one instrument with another, or to relate your observations to theoretical models.

One useful application of **synphot** is to recalculate the value of PHOTFLAM for a given observation using the latest calibration files. For example, to recalculate PHOTFLAM for an FOC observation, you could use the **calcphot** task in **synphot** as follows:

```
sy> calcphot foc,f/96,x96zlrg,f501n ‘unit(1,flam)’ counts
```

The first argument to **calcphot** gives the instrument and its configuration, in this case the FOC *f*/96 camera in full zoomed format with the F501 filter. (See the **obsmode** task in **synphot** and the *Synphot User's Guide* for help with these observation-mode keywords.) The second tells the task to model a flat F_{λ} spectrum having unit flux, and the third tells the task to produce output in units of counts per second. After you run **calcphot**, its `result` parameter will contain the count rate expected from the FOC, given this configuration and spectrum. The PHOTFLAM keyword, defined to be the flux required to produce one count per second, simply equals the reciprocal of this value, which you can print to the screen by typing `=1./calcphot.result` at the IRAF prompt.

Please see the *Synphot User's Guide* for more details on this package, and see appendix A for information on getting the **synphot** dataset, which is not included with STSDAS.

3.4 Displaying HST Spectra

This section shows how to plot your HST spectra for a quick first look and how to generate hardcopies of your plots. Because the STIS data format differs from that of FOS and GHRS, we will discuss STIS data separately.

3.4.1 FOS and GHRS Spectra

Before you work with FOS and GHRS data within STSDAS, you will want to convert the FITS files you received from the Archive into GEIS format (see section 2.3.1 for instructions). After conversion, the `.c1h` file will hold the calibrated flux values for each pixel, the `.c0h` file will hold the corresponding wavelengths, and the `.c2h` file will hold the propagated statistical errors.

Each group of an FOS or GHRS GEIS file contains the results of a separate subintegration. FOS readouts taken in ACCUM mode are cumulative, so the last group contains the results of the entire integration. In contrast, GHRS readouts and FOS readouts in RAPID mode are independent. If you want to see the results of an entire GHRS FP-SPLIT integration, you will need to align and coadd the spectra in the groups of the GHRS file. You can also combine all the groups in an FOS or GHRS data file, without wavelength alignment, using the **rcombine** task in the **hst_calib.ctools** package. See online help for details.

The STSDAS task **sgraph** (in the **graphics.stplot** package) can plot the contents of a single GEIS group. For example, if you want to see group 19 of the calibrated FOS spectrum with rootname `y3bl0104t` you can type

```
st> sgraph y3bl0104t.c1h[19]
```

Given an input flux image (`.c1h`), the task **fwplot** (in the **hst_calib.ctools** package) will look for the corresponding wavelength (`.c0h`) file and plot flux versus wavelength. If requested, it will also look for the error (`.c2h`) file and plot the error bars. To see a plot of the same spectrum as above, but with a wavelength scale and error bars, type

```
st> fwplot y3bl0104t.c1h[19] plterr+
```

If you ever need to plot the contents of multiple groups offset from one another on the same graph, you can use the **grspec** task in the **graphics.stplot** package. For example, to plot groups 1, 10, and 19 of a given flux file, you can type

```
st> grspec y3bl0104t.c1h 1,10,19
```

Note that **grspec** expects group numbers to be listed as a separate parameter, rather than enclosed in the standard square brackets.

3.4.2 STIS Spectra

STIS data files retrieved from the Archive can contain spectra in two different forms: as long-slit spectral images in FITS IMAGE extensions or as extracted echelle spectra in FITS BINTABLE extensions.

You can use **sgraph** to plot STIS long-slit spectra by specifying the image section that contains the spectrum. For example, to plot the entire *x* range of the calibrated two-dimensional spectrum in the first extension of the file `o43ba1bnm_x2d.fits`, averaging rows 100 through 1000, you would type

```
st> sgraph o43ba1bnm_x2d.fits[1][*,100:1000]
```

Displaying the long-slit spectral image using the **display** task (see section 3.2.1 in the HST Introduction) allows you to see the range of your spectrum in *x* and *y* pixel space, so you can choose a suitable image section for plotting.

To plot STIS spectra in BINTABLE extensions, you first need to understand how STIS spectra are stored as binary arrays in FITS table cells. Chapter 2 (section 2.2.2) discusses this format and describes the *selectors* syntax used to specify these data arrays. Each row of a STIS echelle table contains a separate spectral order, and each column contains data of a certain type, such as WAVELENGTH data or FLUX data. To specify a particular array, you must first type the file name, then the extension containing the BINTABLE, then the column selector, then the row selector. For example, to select the WAVELENGTH array corresponding to spectral order 80 of the echelle spectrum in extension 4 of `stis.fits`, you would specify the file as:

```
stis.fits[4][c:WAVELENGTH][r:sporder=80]
```

The **sgraph** task and the **igi** plotting package, to be discussed below, both understand the *selectors* syntax. In particular, if you wanted to plot the flux versus wavelength in STIS echelle order 80, you could type:

```
st> sgraph “stis.fits[4][r:sporder=80] WAVELENGTH FLUX”
```

Remember to include the quotation marks. Otherwise, **sgraph** will complain about too many positional arguments. Note also that **sgraph** understands only row selector syntax; columns are chosen by name.

The STIS-specific **echplot** task is particularly useful for browsing STIS echelle spectra. It can plot single spectral orders, overplot multiple orders on a single plot, or plot up to four orders in separate panels on the same page. For example, to overplot the orders contained in rows two through four and row six on a single page:

```
cl> echplot "stis_x1d.fits[1][r:row=(2:4,6)]" output.igi \
>>> plot_style=m
```

Note that the `plot_style` parameter governs how the spectral orders are plotted. The `plot_style` values `s`, `m`, and `p` plot one order per page, several orders on a single plot, and one order per panel, respectively. The default brightness unit is calibrated FLUX, although you can specify other quantities (e.g., NET counts) using the `flux_col` parameter. See the online help for details.

3.4.3 Producing Hardcopy

This section shows how to generate hardcopies of plots directly and describes **igi**, the Interactive Graphics Interpreter available in STSDAS.

Direct Hardcopies

To print a quick copy of the displayed plot:

1. Type `=gcur` in the command window (where your CL prompt is located).
2. Move the cursor to any location in the graphics window.
3. Press = to write the plot to the graphics buffer.
4. Type `q` to exit graphics mode.
5. At the cl prompt, type `gflush`.

Plots will be printed on the printer defined by the IRAF environment variable stdplot. Type show stdplot to see the current default printer; use set stdplot = printer_name to set the default printer.

The PostScript kernel **psikern** allows you to create PostScript files of your IRAF/STSDAS plots. For example, setting the `device` parameter in a plotting task equal to `psi_port` or `psi_land` invokes **psikern** and directs your plot to either a portrait-mode or a landscape mode PostScript file. For example:

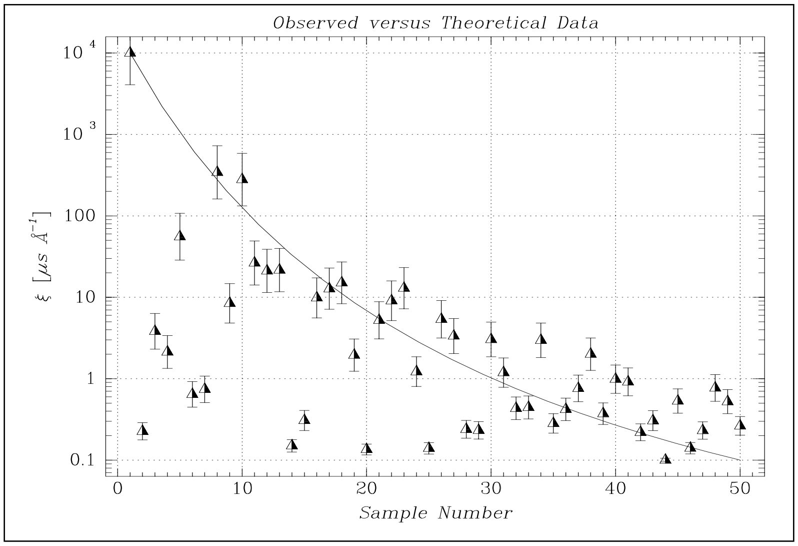
```
st> fwplot y3bl0104t.c1h[19] device=psi_land
st> gflush
/tmp/pskxxxx
```

The above commands would write a plot of flux vs. wavelength in landscape-mode into a temporary PostScript file, named `/tmp/psk`*xxxx* by a UNIX system. See the online help for more about **psikern**, including plotting in color and incorporating PostScript fonts into your plots.

igi

As your plotting needs grow more sophisticated—and especially as you try preparing presentations or publication-quality plots—you should investigate the Interactive Graphics Interpreter, or **igi**. This task, in the STSDAS **stplot** package, can be used with images as well as two- and three-dimensional tables and can draw axes, error bars, labels, and a variety of other features on plots. Different line weights, font styles, and feature shapes are available, enabling you to create complex plots. Figure 3.5 shows a sample plot created in **igi**, however, because **igi** is a complete graphics environment in itself, it is well beyond the scope of this document. You can learn more about **igi** in the *IGI Reference Manual*, available through the *STSDAS Web pages*.

Figure 3.5: Sample igi Plot.



3.5 Analyzing HST Spectra

This section describes some IRAF/STSDAS tasks that can be used for analyzing and manipulating spectral data. Some of these tasks operate directly on HST data files created by the pipeline. However, a number of the most useful IRAF tasks, such as **splot**, require special preparations of data other than STIS two-dimensional spectra. Before discussing these tasks we will first show how to recast your data into forms that are more generally accessible.

3.5.1 Preparing FOS and GHRS Data

The FOS and GHRS data reduction pipelines store fluxes and wavelengths in separate files. In GEIS format, the `.c1h` file contains the flux information and the `.c0h` file contains the wavelength information. Because IRAF tasks generally require both the flux and wavelength information to reside in the same file, you will probably want to create a new file that combines these quantities.

Several options for combining flux and wavelength information are available:

- **resample**: This simple task resamples your flux data onto a linear wavelength scale, creating a new flux file containing the starting wavelength of the new grid in the CRVAL1 keyword and the wavelength increment per pixel in the CD1_1 keyword. Encoding the wavelength information into these standard FITS header keywords makes this format quite portable, but the resampling process loses some of the original flux information. In addition, the error (`.c2h`) and data quality (`.cqh`) files cannot be similarly resampled, limiting the usefulness of this technique.

- **mkmultispec**: This task writes wavelength information into the header of a flux file while preserving all the original information. It is therefore a better choice than **resample** for most applications, and we describe it in more detail below.
- **imtab**: An alternative to writing wavelength information into the header is to use the **imtab** task to create a table recording the wavelength, flux, and if desired, the error data corresponding to each pixel. Many STSDAS tasks, such as those in the STSDAS **fitting** package, can access data in tabular form, so we describe this approach in more detail as well.

mkmultispec

The most convenient method of combining wavelength and flux information, and one that has no effect on the flux data at all, is to use the **mkmultispec** task. This task places wavelength information into the headers of your flux files according to the IRAF multispec-format World Coordinate System (WCS). The multispec coordinate system is intended to be used with spectra having nonlinear dispersions or with images containing multiple spectra, and the format is recognized by many tasks in IRAF V2.10 or later. For a detailed discussion of the multispec WCS, type `help specwcs` at the IRAF prompt.

The **mkmultispec** task can put wavelength information into the flux header files in two different ways. The first involves reading the wavelength data from the `.c0h` file, fitting the wavelength array with a polynomial function, and then storing the derived function coefficients in the flux header file (`.c1h`) in multispec format. Legendre, Chebyshev, or cubic spline (spline3) fitting functions of fourth order or larger produce essentially identical results, all having rms residuals less than 10^{-4} Å, much smaller than the uncertainty of the original wavelength information. Because these fits are so accurate, it is usually unnecessary to run the task in interactive mode to examine them.

If there are discontinuities in the wavelengths, which could arise due to the splicing of different gratings, you should run mkmultispec in interactive mode to verify the fits.

Because mkmultispec can fit only simple types of polynomial functions to wavelength data, this method will not work well with FOS prism data, because of the different functional form of the prism-mode dispersion solution. For prism spectra, use the header table mode of mkmultispec (see below) or create an STSDAS table using imtab.

The other method by which **mkmultispec** can incorporate wavelength information into a flux file is simply to read the wavelength data from the .c0h file and place the entire data array directly into the header of the flux (.c1h) file. This method simply dumps the wavelength value associated with each pixel in the spectrum into the flux header and is selected by setting the parameter `function=table`. To minimize header size, set the parameter `format` to a suitable value. For example, using `format=%8.7g` will retain the original seven digits of precision of the wavelength values, while not consuming too much space in the flux header file.

Be aware that there is a physical limit to the number of header lines that can be used to store the wavelength array (approximately 1000 lines). This limit cannot be overridden. Under ordinary circumstances this limitation is not an issue. However, if many spectral orders have been spliced together, it may not be possible to store the actual wavelength array in the header, and a fit must be done instead

imtab

Another way to combine wavelengths with fluxes is to create an STSDAS table from your spectrum. The **imtab** task in the STSDAS **ttools** package reads a GEIS format spectral image and writes the list of data values to a column of an STSDAS table, creating a new output table if necessary. The following example shows how to create a flux, wavelength, and error table from group eight of a GEIS-format FOS dataset:

```
cl> imtab y0cy0108t.c0h[8] y0cy0108t.tab wavelength
cl> imtab y0cy0108t.c1h[8] y0cy0108t.tab flux
cl> imtab y0cy0108t.c2h[8] y0cy0108t.tab error
```

The last word on each command line labels the three columns “wavelength”, “flux”, and “error”.

Constructing tables is necessary if you plan to use certain tasks—such as those in the STSDAS **fitting** package—that do not currently recognize the multispec format WCS header information. Tabulating your spectra is also the best option if you want to join two or more spectra taken with different gratings into a single spectrum covering the complete wavelength range. Because the data are stored as individual wavelength-flux pairs, you do not need to resample, and therefore degrade the individual spectra to a common, linear dispersion scale before joining them. Instead, you could create separate tables for spectra from different gratings, and then combine the two tables using, for example, the **tmerge** task:

```
cl> tmerge n5548_h13.tab,n5548_h19.tab n5548.tab append
```

Note that you will first have to edit out any regions of overlapping wavelength from one or the other of the input tables so that the output table will be monotonically increasing (or decreasing) in wavelength.

3.5.2 Preparing STIS Spectra for Analysis

Calibrated STIS spectra emerge from the pipeline either as two-dimensional images (`_x2d` files) or as one-dimensional spectra in tabular form (`_x1d` files.) You can analyze calibrated two-dimensional STIS spectra in IRAF as you would with any other long-slit spectral image, because their headers already contain the necessary wavelength information. Tabulated STIS spectra can be analyzed directly using STSDAS tasks that understand the *selectors* syntax described in section 2.2.2. However, to use IRAF tasks, such as **splot**, that rely on the multispec WCS or to use STSDAS tasks that do not understand three-dimensional tables, you will have to prepare your data appropriately. This section describes two useful tasks for putting your data in the proper form:

- **tomultispec**: This task is the STIS analog to **mkmultispec**, described above. It extracts STIS spectra from tables and writes them as IRAF spectral images with wavelength information in the header.
- **txtable**: This task extracts specified data arrays from STIS table cells and places them in conventional two-dimensional tables for easier access.
- **tximage**: Extracts specified data arrays from STIS table cells and places them into 1-D images. This task can write single group GEIS files.

tomultispec

The **tomultispec** task in the **stsdas.hst_calib.ctools** package extracts one or more spectral orders from a STIS table, fits a polynomial dispersion solution to each wavelength array, and stores the spectra in an output file in original IRAF format (OIF), using the multispec WCS. This task is layered upon the **mkmultispec** task, which performs a similar operation for FOS and GHRS calibrated spectra (see "mkmultispec" on page 3-25). Most of the parameters for **tomultispec** echo those for **mkmultispec**. As a helpful navigational aid, the STIS spectral order numbers are written to the corresponding *beam* numbers in the multispec image; the aperture numbers are indexed sequentially starting from one. You can choose to fit the dispersion solution interactively, but the default fourth-order Chebyshev polynomial will likely suffice for all STIS spectral orders, except for prism-dispersed spectra. However, you cannot use the interactive option if you are selecting more than one order from the input file.

For example, if you want to write all spectral orders from the STIS file `myfile_x1d.fits` to a multispec file:

```
cl> tomultispec myfile_x1d.fits new_ms.imh
```

Note that the `.imh` suffix on the output file specifies that the output file is to be an OIF file. This format is similar to GEIS format, in that it consists of two files: a header file (`.imh`) and a binary data file (`.pix`). The output format for **tomultispec** will always be OIF.

If you want to select particular spectral orders, rather than writing all the orders to the multispec file, you will need to use the **selectors** syntax. To select only the spectrum stored in row nine of the input table, the previous example would change to:

```
cl> tomultispec "myfile_x1d.fits[r:row=9]" new_ms.imh
```

Note that the double quote marks around the file name and row selector are necessary to avoid syntax errors. To select a range of rows, say rows nine through eleven:

```
cl> tomultispec "myfile_x1d.fits[r:row=(9:11)]" new_ms.imh
```

You can also select rows based upon values in some other column. For example, to select all rows whose spectral order lies in the range 270 to 272, type:

```
cl> tomultispec "myfile_x1d.fits[r:sporder=(270:272)]" \
>>> new_ms.imh
```

The calibrated flux is extracted by default. However, other intensity data can be specified by setting the `flux_col parameter`.

Be careful not to restrict the search for matching rows too heavily

Column selectors cannot be used with tomultispec

Choose the type of fitting function for the tomultispec dispersion solution with care. Using the table option, which writes the entire wavelength array to the image header for each order, will fail if more than about three orders are selected. This restriction results from a limit to the number of keywords that can be used to store the dispersion relation.

txtable

Tabulated STIS spectra are stored as data arrays within individual cells of FITS binary tables (see section 2.2.2). These tables are effectively three-dimensional, with each column holding a particular type of quantity (e.g., wavelengths, fluxes), each row holding a different spectral order, and each cell holding a one-dimensional array of values spanning the wavelength space of the order. The **txtable** in the **tables.ttools** package extracts these data arrays from the cells specified with the selectors syntax and stores them in the columns of conventional two-dimensional binary tables.

For example, suppose the first extension of the FITS file `data.fits` contains a STIS echelle spectrum and you want to extract only the wavelength and flux arrays corresponding to spectral order 68. You could then type:

```
tt> txtable “data.fits[1][c:WAVELENGTH,FLUX][r:sporder=68]” \
>>> out_table
```

This command would write the wavelength and flux arrays to the columns of the output table `out_table`. To specify multiple rows in a tabulated echelle spectrum, you would type:

```
tt> txtable “data.fits[1][c:WAVELENGTH,FLUX][r:row=(10:12)]” \
>>> echl
```

This command would generate three separate output files named `echl_r0010.tab`, `echl_r0011.tab`, and `echl_r0012.tab`.

See the online help for more details on **txtable** and the selectors syntax, and remember to include the double quotation marks.

The similar **tximage** task can be used to generate single-group GEIS files from STIS data, which can then be used as input to tasks such as **resample**.

```
tt> tximage "data.fits[1][c:WAVELENGTH][r:row=4]" wave.hhh
tt> tximage "data.fits[1][c:FLUX][r:row=4]" flux.hhh
```

3.5.3 General Tasks for Spectra

IRAF has many tasks for analyzing both one- and two-dimensional spectral data. Many observers will already be familiar with **noao.onedspec** and **noao.twodspec** packages, and those who are not should consult the online help. Table 3.5 lists some of the more commonly used IRAF/STSDAS spectral analysis tasks, and below we briefly describe **splot**, one of the most versatile and useful. Remember that many of these tasks expect to find WCS wavelength information in the header, so you should first run **mkmultispec** or **tomultispec** on your data, if necessary.

Table 3.5: Tasks for Working with Spectra

Task	Package	Input Format	Purpose
boxcar	images.imfilter	Image	Boxcar smooth a list of images
bplot	noao.onedspec	Multispec image	Plot spectra non-interactively
continuum	noao.onedspec	Image	Continuum normalize spectra
fitprofs	noao.onedspec	Image	Non-interactive Gaussian profile fitting to features in spectra and image lines
gcopy	stsdas.toolbox.imgtools	GEIS image	Copy multigroup images
grlist	stsdas.graphics.stplot	GEIS image	List file names for all groups in a GEIS image; used to make lists for tasks that do not use group syntax
grplot	stsdas.graphics.stplot	GEIS image	Plot arbitrary lines from 1-D image; overplots multiple GEIS groups; no error or wavelength information is used
grspec	stsdas.graphics.stplot	Multispec GEIS image	Plot arbitrary lines from 1-D image; stack GEIS groups
magnify	images.imgeom	Image	Interpolate spectrum on finer (or coarser) pixel scale
nfit1d	stsdas.analysis.fitting	Image, table	Interactive 1-D non-linear curve fitting (see section 3.5.4)
ngaussfit	stsdas.analysis.fitting	Image, table	Interactive 1-D multiple Gaussian fitting (see section 3.5.4)
poffsets	stsdas.hst_calib.ctools	GEIS image	Determine pixel offsets between shifted spectra
rapidlook	stsdas.hst_calib.ctools	GEIS image	Create and display a 2-D image of stacked 1-D images
rcombine	stsdas.hst_calib.ctools	GEIS image	Combine (sum or average) GEIS groups in a 1-D image with option of propagating errors and data quality values
resample	stsdas.hst_calib.ctools	GEIS image	Resample FOS and GHRS data to a linear wavelength scale (see section 3.5.1)
sarith	noao.onedspec	Multispec image	Spectrum arithmetic
scombine	noao.onedspec	Multispec image	Combine spectra
sfit	noao.onedspec	Multispec image	Fit spectra with polynomial function
sgraph	stsdas.graphics.stplot	Image, table	Plot spectra and image lines; allows overplotting of error bars and access to wavelength array (see section 3.4.1)
specalign	stsdas.hst_calib.ctools	GEIS image	Align and combine shifted spectra (see **poffsets**)
specplot	noao.onedspec	Multispec image	Stack and plot multiple spectra
splot	noao.onedspec	Multispec image	Plot and analyze spectra & image lines (see “splot” on page 3-31)

splot

The **splot** task in the IRAF **noao.onedspec** package is a good general analysis tool that can be used to examine, smooth, fit, and perform simple arithmetic operations on spectra. Because it looks in the header for WCS wavelength information, your file must be suitably prepared. Like all IRAF tasks, **splot** can work on only one group at a time from a multigroup GEIS file. You can specify which GEIS group you want to operate on by using the square bracket notation, for example:

```
cl> splot y0cy0108t.c1h[8]
```

If you don’t specify a group in brackets, **splot** will assume you want the first group. In order to use **splot** to analyze your FOS or GHRS spectrum, you will first need to write the wavelength information from your `.c0h` file to the header of your `.c1h` files in WCS, using the **mkmultispec** task (see “mkmultispec” on page 3-25).

The **splot** task has *many* available options described in detail in the online help. Table 3.6 summarizes a few of the more useful cursor commands for quick reference. When you are using **splot**, a log file saves results produced by the equivalent width or de-blending functions. To specify a file name for this log file, you can set the `save_file` parameter by typing, for example:

```
cl> splot y0cy0108t.c1h[8] save_file=results.log
```

If you have used **tomultispec** to transform a STIS echelle spectrum into `.imh/.pix` OIF files with WCS wavelength information (see “tomultispec” on page 3-27), you can step through the spectral orders stored in image lines using the “)”, “(“, and “#” keys. To start with the first entry in your OIF file, type:

```
cl> splot new_ms.imh 1
```

You can then switch to any order for analysis using the ")" key to increment the line number, the "(" key to decrement, and the “#” key to switch to a specified image line. Note the beam label that gives the spectral order cannot be used for navigation. See the online help for details.

Table 3.6: Useful splot Cursor Commands

Command	Purpose
Manipulating spectra	
f	Arithmetic mode; add and subtract spectra
l	Convert spectrum from f_ν to f_λ (invert transformation with “n”)
n	Convert spectrum from f_λ to f_ν
s	Smooth with a boxcar
u	Define linear wavelength scale using two cursor markings
Fitting spectra	
d	Mark two continuum points & de-blend multiple Gaussian line profiles
e	Measure equivalent width by marking points around target line
h	Measure equivalent width assuming Gaussian profile
k	Mark two continuum points and fit a single Gaussian line profile
m	Compute the mean, RMS, and S/N over marked region
t	Enter interactive curve fit function (usually used for continuum fitting)
Displaying and redrawing spectra	
a	Expand and autoscale data range between cursor positions
b	Set plot base level to zero
c	Clear all windowing and redraw full current spectrum
r	Redraw spectrum with current windowing
w	Window the graph
x	Etch-a-sketch mode; connects two cursor positions
y	Overplot standard star values from calibration file
z	Zoom graph by a factor of two in X direction
$	Switch between physical pixel coordinates and world coordinates
General file manipulation commands	
?	Display help
g	Get another spectrum
i	Write current spectrum to new or existing image
q	Quit and go on to next input spectrum

3.5.4 STSDAS fitting Package

The STSDAS **fitting** package contains several tasks, as listed in table 3.7, for fitting and analyzing spectra and images. The **ngaussfit** and **nfit1d** tasks, in particular, are very good for interactively fitting multiple Gaussians and nonlinear functions, respectively, to spectral data. These tasks do not currently recognize the multispec WCS method of storing wavelength information. They recognize the simple sets of dispersion keywords such as W0, WPC and CRPIX, CRVAL, and CDELT, but these forms apply only to linear coordinate systems and therefore would require resampling of your data onto a linear wavelength scale first. However, these tasks do accept input from STSDAS tables, in which you can store the wavelength and flux data value pairs or wavelength, flux, error value triples (see "imtab" on page 3-26).

Table 3.7: Tasks in the STSDAS fitting Package

Task	Purpose
function	Generate functions as images, tables, or lists
gfit1d	Interactive 1-d linear curve fit to images, tables, or lists
i2gaussfit	Iterative 2-d Gaussian fit to noisy images (script)
nfit1d	Interactive 1-d non-linear curve fit to images, tables, or lists
ngaussfit	Interactive 1-d multiple Gaussian fit to images, tables, or lists
n2gaussfit	2-d Gaussian fit to images
prfit	Print contents of fit tables created by fitting task

When using tasks such as **ngaussfit** and **nfit1d**, you must provide initial guesses for the function coefficients as input to the fitting algorithms. You can either specify these initial guesses via parameter settings in the task's parameter sets (psets) or enter them interactively. For example, suppose you want to fit several features using the **ngaussfit** task. Using the default parameter settings, you can start the task by typing:

```
fi> ngaussfit n4449.hhh linefits.tab
```

This command reads spectral data from the image `n4449.hhh` and stores the results of the line fits in the STSDAS table `linefits.tab`. After you start the task, your spectrum should appear in a plot window and the task will be left in cursor input mode. You can use the standard IRAF cursor mode commands to rewindow the plot, resticting your display to the region around a particular feature or features that you want to fit. You may then want to:

- Define a sample region (using the cursor mode S command) over which the fit will be computed so that the task will not try to fit the entire spectrum.
- Define an initial guess for the baseline coefficients by placing the cursor at two baseline locations (one on either side of the feature to be fitted) using the B keystroke.
- Use the R keystroke to redraw the screen and see the baseline that you've just defined.
- Set the initial guesses for the Gaussian centers and heights by placing the cursor at the peak of each feature and typing P.
- Press F to compute the fit once you've marked all the features you want to fit.

The results will automatically be displayed. You can use the `:show` command to see the coefficient values.

Note that when the **ngaussfit** task is used in this way (i.e., starting with all default values), the initial guess for the FWHM of the features will be set to a value of one. Furthermore, this coefficient and the coefficients defining the baseline are held fixed by default during the computation of the fit, unless you explicitly tell the task through cursor *colon* commands[3] to allow these coefficients to vary. It is sometimes best to leave these coefficients fixed during an initial fit, and then to allow them to vary during a second iteration. This rule of thumb also applies to the setting of the `errors` parameter which controls whether or not the task will estimate error values for the derived coefficients. Because the process of error estimation is very CPU-intensive, it is most efficient to leave the error estimation turned off until you've got a good fit, and then turn the error estimation on for one last iteration.

Figure 3.6 and figure 3.7 shows the results of fitting the Hβ (4861Å) and [OIII] (4959 and 5007 Å) emission features in the spectrum of NGC 4449. The resulting coefficients and error estimates (in parentheses) are shown in figure 3.7.

[3] To see the online help for details and a complete listing of cursor mode colon commands: type `help cursor`.

Figure 3.6: Fitting Hβ and [OIII] Emission Features in NGC 4449

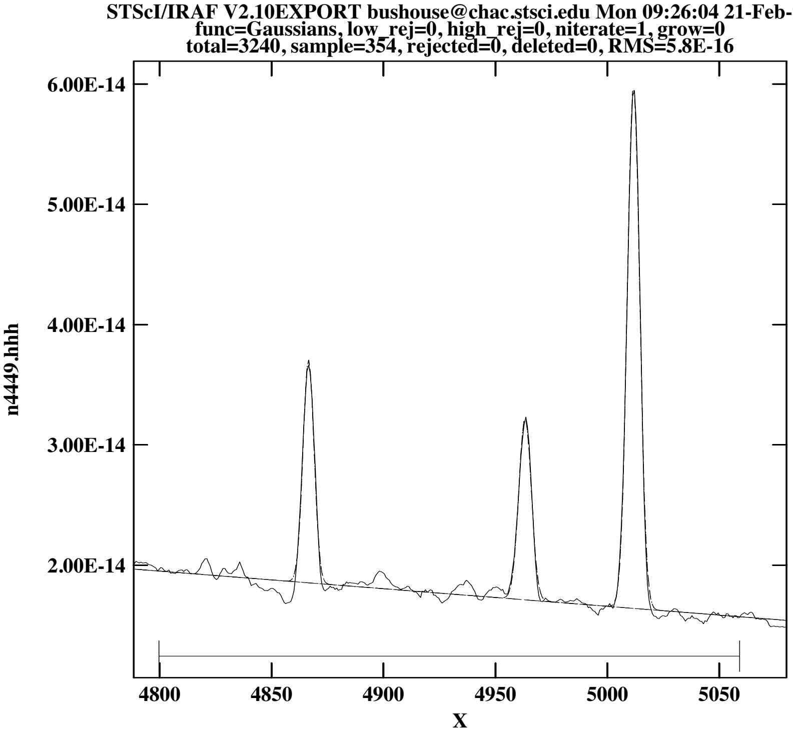


Figure 3.7: Coefficients and Error Estimates

```
function = Gaussians
coeff1 = 8.838438E-14    (0.)             - Baseline zeropoint (fix)
coeff2 = -1.435682E-17   (0.)             - Baseline slope (fix)
coeff3 = 1.854658E-14    (2.513048E-16)    - Feature 1: amplitude (var)
coeff4 = 4866.511        (0.03789007)     - Feature 1: center (var)
coeff5 = 5.725897        (0.0905327)      - Feature 1: FWHM (var)
coeff6 = 1.516265E-14    (2.740680E-16)    - Feature 2: amplitude (var)
coeff7 = 4963.262        (0.06048062)     - Feature 2: center (var)
coeff8 = 6.448922        (0.116878)       - Feature 2: FWHM (var)
coeff9 = 4.350271E-14    (2.903318E-16)     - Feature 3: amplitude (var)
coeff10 = 5011.731       (0.01856957)     - Feature 3: center (var)
coeff11 = 6.415922       (0.03769293)     - Feature 3: FWHM (var)
rms           = 5.837914E-16
grow          = 0.
naverage      = 1
low_reject    = 0.
high_reject   = 0.
niterate      = 1
sample        =  4800.132:5061.308
```

3.5.5 specfit

The **specfit** task, in the STSDAS **contrib** package, is another powerful interactive facility for fitting a wide variety of emission-line, absorption-line, and continuum models to a spectrum. This task was written by Gerard Kriss. Extensive online help is available to guide you through the task,[4] although because it is a contributed task, little to no support is provided by the STSDAS group.

The input spectrum to **specfit** can be either an IRAF image file or an ASCII file with a simple three-column (wavelength, flux, and error) format. If the input file is an IRAF image, the wavelength scale is set using values of W0 and WPC or CRVAL1 and CDELT1. Hence, for image input, the spectral data must be on a linear wavelength scale. In order to retain data on a non-linear wavelength scale, it is necessary to provide the input spectrum in an ASCII file, so that you can explicitly specify the wavelength values associated with each data value. The online help explains a few pieces of additional information that must be included as header lines in an input text file.

By selecting a combination of functional forms for various components, you can fit complex spectra with multiple continuum components, blended emission and absorption lines, absorption edges, and extinction. Available functional forms include linear, power-law, broken power-law, blackbody, and optically thin recombination continua, various forms of Gaussian emission and absorption lines, absorption-edge models, Lorentzian line profiles, damped absorption-line profiles, and mean galactic extinction.

3.6 References

3.6.1 Available from STScI

(http://stsdas.stsci.edu/STSDAS.html

- *STSDAS Users Guide* , version 1.3, September 1994.
- *STSDAS Installation and Site Managers Guide*, version 2.3, June 2001.
- *Synphot Users Guide*, December 1998.
- *IGI Reference Manual* , version 1.3, October 1992.

[4] Additional information is available in the *Astronomical Data Analysis Software and Systems III*, ASP Conference Series, Vol. 61, page 437, 1994.

3.6.2 Available from NOAO

(http://iraf.noao.edu/docs/docmain.html)

- *A Beginners Guide to Using IRAF,* 1993, J. Barnes.
- *Photometry Using IRAF*, 1994, L. Wells.
- *A User's Guide to Stellar CCD Photometry with IRAF*, 1992, P. Massey and L. Davis.

3.6.3 Other References Cited in This Chapter

- Horne, K., 1988, in *New Directions in Spectrophotometry*, A.G.D. Philip, D.S. Hayes, and S.J. Adelman, eds., L. Davis Press, Schenectady NY, p. 145.
- Koorneef, J., R. Bohlin, R. Buser, K. Horne, and D. Turnshek, 1986, in *Highlights of Astronomy*, Vol. 7, J.-P. Swinds, ed., Reidel, Dordrecht, p. 833.
- Kriss, G., 1994, in *Astronomical Data Analysis Software and Systems III* , PASP Conference Series, Vol. 61, p. 437.

PART II:

NICMOS Data Handbook

This handbook is designed to help you manipulate, process and analyze data from the Near-Infrared Camera and MultiObject Spectrograph (NICMOS) on board the Hubble Space Telescope (HST).

NICMOS Introduction

How to Use this Handbook

This handbook is designed to help you manipulate, process and analyze data from the Near-Infrared Camera and MultiObject Spectrograph (NICMOS) on board the Hubble Space Telescope (HST). This is presented as an independent and self-contained document, extensively built on the contents of an older edition (version 3.0) of the HST Data Handbook. The old HST Data Handbook has now been subdivided into separate volumes for each instrument. Users who wish to find more general information about details of acquiring HST data from archive, their file formats, and general purpose software for displaying and processing these data, are referred to a companion volume, the Introduction to the HST Data Handbook.

The current edition of the NICMOS Data Handbook is being written in late 2001, as we approach servicing mission 3b (SM3b) and the installation of the NICMOS Cooling System (NCS). Therefore, the material presented in this volume are based on the experience gained in NICMOS during Cycles 7 and 7N, before the cryogen was exhausted. When NICMOS is revived with NCS, undoubtedly there will be differences in its performance and characteristics. At the very least, it is expected to operate at a warmer temperature. This will lead to differences in calibration in Cycle 11 and beyond, as many of the instrument properties are temperature sensitive (e.g., dark current, bias behavior, quantum efficiency and hence, photometric calibration). Moreover, the NCS itself may introduce other changes in the way the NICMOS data are calibrated and analyzed, depending on the temperature stability it achieves. Another significant change from previous cycles is the difference in NICMOS commanding and operations, which includes automatically obtaining dark frames after each HST passage through the South Atlantic Anomaly (SAA).

While this handbook provides comprehensive information for treatment of the NICMOS data obtained during cycles 7 and 7N, most of the material presented here are also applicable for observations in Cycle 11 and beyond. Sometime after SMOV3b, and once the in-orbit performance of the NICMOS/NCS is well understood, we expect to provide a major update of this handbook.

In the meantime, readers are advised to consult the NICMOS web pages on the Space Telescope Science Institute web site for the latest information regarding NICMOS performance and calibration.

Bahram Mobasher, Chief Editor, HST Data Handbook
Mark Dickinson, Editor, NICMOS Data Handbook

CHAPTER 1:

Instrument Overview

In this chapter. . .

This chapter presents a brief overview of the Near Infrared Camera and Multi-Object Spectrometer (NICMOS), its capabilities, readout modes, and data products.

1.1 Instrument Overview

NICMOS was built by Ball Aerospace Corporation for the University of Arizona, under the direction of Rodger I. Thompson, the Principal Investigator. A basic description of the instrument and its on-orbit performance through the Servicing Mission Orbital Verification program is provided by Thompson et. al (1998).[1] We encourage all NICMOS users to reference this paper and to review the related papers in the special issue of *ApJ Letters* which describe the Early Release Observations and demonstrate the scientific capabilities of NICMOS. Many additional papers have now appeared in the literature which document the use of NICMOS in a wide variety of scientific applications. The *NICMOS Instrument Handbook* and the NICMOS WWW pages at STScI are also valuable sources of information for the NICMOS user, particularly concerning technical details of the instrument, as well as a history of its performance throughout its lifetime. The proceedings to the *1997 HST Calibration*

[1] Thompson, R.I., M. Rieke, G. Schneider, D.C. Hines, and M.R. Corbin, 1998, *ApJL*, 492, L95.

Workshop (eds. S. Casertano, R. Jedrzejewski, T. Keyes & M. Stevens, STScI publication) also include useful information about NICMOS and its calibration and performance, although some of that information is now superseded by material included in this handbook and other, more recent documents. Finally, the NICMOS edition of the *Space Telescope Analysis Newsletter (STAN)*, which is periodically distributed by e-mail[2], provides regular notices and updates to information about NICMOS and NICMOS data reduction. Back issues of the NICMOS STAN are archived at the STScI NICMOS web site.

The first phase of life for NICMOS took place during HST Cycle 7, with a special, supplemental call for proposals issued as Cycle 7N. Its cryogens were exhausted in January 1999, and the instrument was deactivated and subsequently decommissioned. In HST Cycle 11, the installation of the NICMOS Cooling System (NCS) is expected to bring NICMOS back down to cryogenic temperatures, and thus return it to regular service. Its performance with the NCS is, naturally, unknown at this time, although predictions are presented in the most recent edition of the *NICMOS Instrument Handbook*.

This edition of the NICMOS Data Handbook is based on experience with NICMOS data obtained during the first two years of operation, i.e., HST Cycles 7 and 7N, before cryogen exhaustion. It is expected that instrument performance may be somewhat different in Cycle 11 and beyond with the NICMOS Cooling System, and that some aspects of NICMOS data reduction will accordingly require revision. NICMOS users should carefully monitor developments and updates posted on the STScI NICMOS web pages when analyzing data taken in Cycle 11 and beyond.

NICMOS provides imaging capabilities in broad, medium, and narrow band filters, broad-band imaging polarimetry, coronographic imaging, and slitless grism spectroscopy, in the wavelength range 0.8–2.5 µm. NICMOS is an axial instrument and has three adjacent but not contiguous cameras, designed to operate independently and simultaneously. Each camera has a different magnification scale, and is equipped with a dedicated 256 × 256 HgCdTe Rockwell array. The approximate pixel sizes and fields of view are 0″.043 and 11″ × 11″ for Camera 1 (referred to as NIC1), 0″.075 and 19″.2 × 19″.2 for NIC2, and 0″.2 and 51″.2 × 51″.2 for NIC3.

Each camera is provided with its own set of filters, mounted on three independent wheels. There are a total of 20 filter positions on each wheel,

[2] To subscribe to the *STAN*, send a message to majordomo@stsci.edu with the Subject: line blank and “subscribe nicmos_news” in the body).

of which one is blank (i.e., a cold, opaque filter used in lieu of a dark slide). Three of the other positions are occupied by either polarizers or grisms. The remaining 16 positions of each filter wheel are occupied by broad, medium, and narrow band filters. The list of these filters is given in the *NICMOS Instrument Handbook*. The filters (including polarizers and grisms) cannot be crossed with each other, and are used as single optical elements.

NIC1 and NIC2 each contain three polarizers, whose principal axes of transmission are separated by approximately 120 degrees (for the exact polarizer orientations and other details, see section 5.7 of this manual, and also chapter 5 of the *NICMOS Instrument Handbook*). The spectral coverage is fixed for each camera. The polarizers cover the wavelength range 0.8–1.3 μm in NIC1, and 1.9-2.1 μm in NIC2. Observations in the three polarizers of each camera are used to derive the Stokes parameters of linearly polarized light.

The filter wheel of NIC3 contains three grisms which can be used to perform slitless spectroscopy in the wavelength range 0.8–2.5 μm. The three grisms cover the range 0.8–1.2 μm, 1.1–1.9 μm, and 1.4–2.5 μm, respectively.

In NIC2, a coronographic spot is imaged onto the focal plane and provides a circular occulted region 0″.3 in radius (with a useful effective radius of 0″.4). For coronographic imaging, an acquisition sequence is required at the beginning of the observation to center the target under the occulting spot.

Each 256 × 256 detector array is divided into four 128 × 128 quadrants, each of which is read out by an amplifier at the corner of the quadrant. There are four amplifiers in each camera. Unlike CCDs, infrared array pixels are read independently; problems like charge transfer efficiency or bleeding are not present. The three cameras operate independently; optical elements, integration times, and readout modes can be different in each.

1.2 Detector Readout Modes

NICMOS does not have a physical shutter mechanism, and exposures are obtained through a sequence of reset and read operations. In particular, a typical exposure will be the product of the following steps:

1. ***Array reset***: the pixels are set to the bias level.
2. ***Array read***: the charge in each pixel is measured and stored in the on-board computer's memory. This read is performed immediately after the reset, and contains the reference level for the exposure (*zeroth read*). In practice, the readout is performed 0.203 seconds after the reset, implying that it represents a finite, though very short, exposure. This readout is performed non-destructively; the charge in each pixel is left intact.
3. ***Integration***: NICMOS integrates for the user-specified time.
4. ***Array read***: the charge in each pixel is measured and stored in the on-board computer's memory. Again, the readout is non-destructive.

The beginning of an integration is marked by the zeroth read, which is always preceded by a reset. Since all readouts are non-destructive, i.e., do not change the value of the charge accumulated on the pixel, the last two steps of the sequence above can be repeated multiple times, and the last read of the sequence will be called the *final read*. The total integration time of an exposure is defined as the time between the final and the zeroth read of the first pixel in the array. The scientific image is given by the difference between the final and the zeroth readouts.

Four readout modes have been defined for NICMOS, exploiting the flexibility allowed by the non-destructive reads:

- MULTIACCUM
- ACCUM
- BRIGHTOBJ
- RAMP (not available in Cycle 11 and beyond).

Each mode is described in the following sections, with larger emphasis on MULTIACCUM, which is by far the most used and best calibrated mode. RAMP mode was never used for on-orbit science observations during Cycle 7, and will no longer be implemented for use in Cycle 11 and beyond.

1.2.1 MULTIACCUM

In a MULTIACCUM (MULTIple ACCUMulate) exposure, the zeroth read is followed by several other non-destructive readouts during the course of a single integration. All of the readouts are stored in the on-board computer's memory and sent to the ground. Because the readouts are non-destructive, accumulated counts are built up from one readout to the next, with the last readout containing the accumulated counts from the entire integration time of the observation. In an exposure, the number of readouts after the zeroth and the temporal spacing between each read is selected by the user from a set of 16 pre-defined *sequences*. The sequence chosen by the user is stored in the value of the `SAMP_SEQ` keyword in the science data files. The user specifies the number of readouts through the `NSAMP` keyword during the Phase II proposal process. `NSAMP+1` (including the zeroth read) images will be returned to the ground. For NICMOS the maximum value of `NSAMP` is 25 in each sequence (for a total of 26 images returned to the ground).

Because MULTIACCUM gives information not only at the beginning and at the end of an exposure, but also at intermediate times, it is the mode of choice for the vast majority of astronomical observations, from objects with large dynamical range to deep field integrations. The intermediate reads can also be used to remove the effects of cosmic ray hits and of saturated pixels from the final processed image.

The images returned to the ground by the MULTIACCUM readout are raw detector readouts, since not even the bias level (the zeroth read) is subtracted. This operation is performed by the ground calibration pipeline.

1.2.2 ACCUM

ACCUM is a simplified version of MULTIACCUM: the zeroth read is followed by one read (the final readout) after an amount of time specified by the integration time. The difference between the final and the zeroth readouts is computed on-board, and the resulting image is sent to the ground. In this form, the ACCUM mode produces data very similar to the more familiar CCD images. A variation to this basic operation is available, which replaces the single initial and final readouts with multiple (initial and final) readouts. After the initial reset pass, *n* non-destructive reads of the detector immediately follow, as close together in time as allowed by the detector electronics. The average of the *n* values is stored as the initial value for each pixel. At the end of the integration, there are again *n* non-destructive readouts with the final value for each pixel being the average of the *n* reads. The number *n* of initial and final reads is specified by the observer and is recorded in the value of the NREAD (number of reads) header keyword in the science data files. The returned image is the difference between the averaged final and initial values. The integration time is defined as the time between the first read of the first pixel in the initial *n* passes and the first read of the first pixel in the final *n* passes. The advantage of the multiple initial and final (MIF) readout method is that, in theory, the read noise associated with the initial and final reads should be reduced by a factor of $\sqrt{n}$, where *n* is the number of reads (see, e.g., Fowler and Gatley 1990, ApJ, 353, L33). In practice, actual noise reduction in NICMOS observations is generally rather less than $\sqrt{n}$ for a variety of reasons, such as amplifier glow (see section 4.1.1). The supported NREAD values are 1 and 9.

1.2.3 BRIGHTOBJ

The BRIGHTOBJ (BRIGHT OBJect) mode provides a way to observe objects that would usually saturate the detector in less than the minimum available exposure time (which is the amount of time it takes to read out the array and is 0.203 seconds). In BRIGHTOBJ mode each individual pixel (per quadrant) is successively reset, read, integrated for a time requested by the observer, and read again, and then these steps are performed for the next pixel in the quadrant. The returned image contains the number of counts accumulated between the initial and final reads for each pixel (just like ACCUM mode). Since each quadrant contains 16,384 pixels, the total elapsed time to take an image in this mode is 16,384 times the requested exposure time for each pixel.

BRIGHTOBJ mode was rarely used for on-orbit science observations and is essentially uncalibrated. In Cycle 11, BRIGHTOBJ is an available observing mode for the special case of acquisition of very bright targets for coronagraphy, but is not supported by STScI.

1.2.4 RAMP

RAMP mode was designed to use multiple non-destructive reads during the course of a single exposure much like MULTIACCUM, but only a single image is sent to the ground. Although RAMP mode could, in principle, be used to obtain the benefits of a MULTIACCUM exposure without the large data volume, the difficulty of implementing infallible algorithms for on-board processing made this mode impractical, and its use for on-orbit observations was discouraged in Cycles 7 and 7N. Indeed, almost no data were obtained in RAMP mode, and it will not be supported for NICMOS observing in Cycle 11 and beyond. We describe the mode here for historical reasons only, but it will not be considered otherwise in this *Handbook*.

In RAMP mode, the total integration time T is divided into n equal intervals $t = T / n$. Each readout is differenced (on-board) with the previous readout and used to compute a running mean of the number of counts (per sample interval) and an associated variance for each pixel. Large deviations from the running mean are used to detect saturation or a cosmic ray hit. At the end of the exposure, the data sent to the ground comprise a mean countrate image, plus the variance and the number of valid samples used to compute each pixel value. The effective exposure time for the returned image is the sample interval t.

Almost no data were obtained in RAMP mode during Cycles 7 and 7N, and it will not be supported for NICMOS observing in Cycle 11 and beyond. The reduction and analysis of RAMP mode data will not be discussed further in this Handbook.

CHAPTER 2:

Data Structures

In this chapter. . .

This chapter is a guide to the structure of NICMOS data. The data file naming convention, formats, and organization are described, as are the file header keywords. The connection between the Phase II exposure logsheets and the data you receive is also explained, together with the available paper products.

2.1 NICMOS Data Files

STScI automatically processes and calibrates all NICMOS data and archives the data files resulting from pipeline processing in FITS format. If you have retrieved NICMOS files from the Archive, you will notice that their names look like this:

```
n3w2a1wqm_cal.fits
```

The first part of the file name (`n3w2a1wqm`) is the *rootname*, identifying the dataset to which the file belongs, the second (`cal`) is the *suffix*, identifying the type of data the file contains, and the third (`fits`) indicates that this is a FITS format file. Chapter 2 of the HST Introduction shows how to access the data contained in NICMOS FITS files, while appendix B of that volume explains how to decipher the rootnames of these files and explains why some of them are grouped into data *associations*. This section describes the files that constitute a NICMOS dataset.

2.1.1 File Name Suffixes

Each file in a NICMOS dataset has a three-character suffix that uniquely identifies the file contents. The file name suffixes for NICMOS and the corresponding file contents are summarized below. The files that contain final calibrated data (which you will most likely use for analysis) are highlighted in bold italics and are `_cal` and `_mos` for unassociated and associated data, respectively. This list contains *all* of the files that the pipeline *can* produce. For some observing strategies not all of the processing steps are performed and only a subset of these files will be produced by the pipeline.

- Raw Data Files
  - *Raw Science File* (`_raw`):
    This FITS file contains the raw image data received from the spacecraft. One file per exposure is created (a MULTIACCUM exposure is considered a single exposure irrespective of the number of samples specified).
  - *Support File* (`_spt`):
    This FITS file contains supporting information about the observation, the spacecraft telemetry and engineering data from the instrument that was recorded at the time of the observation, including detector temperature measurements.
  - *Association Table* (`_asn`):
    This file is a FITS binary table that contains the list of datasets making up an association.
- Calibrated Data Files
  - *Calibrated Science File* (`_cal`):
    This FITS file contains the calibrated science data for an *individual dataset*, and is produced by the pipeline calibration task **calnica** (see chapter 3). The input to **calnica** are the `_raw` images. For a MULTIACCUM exposure, this file contains a single science image formed by combining the data from all samples.
  - *Intermediate Multiaccum Science File* (`_ima`):
    This FITS file is also produced by the pipeline task **calnica** and contains the calibrated science data for all samples of a MULTIACCUM dataset before the process of combining the individual readouts into a single image has occurred. This file is only produced for MULTIACCUM observations.

- *Mosaic Files* (_mos):
  These FITS files contain the composite target and, for chopped pattern sequences, background region images constructed by the pipeline task **calnicb** for an associated set of observations (see chapter 3). The input to **calnicb** are the calibrated _cal images from **calnica** and the _asn association table. Target images are co-added and background-subtracted. The value of the last character of the rootname is 0 for targets, and 1 to 8 for background images. These files are only produced for an associated set of observations.
- *Post-calibration Association Table* (_asc):
  This table is produced by the pipeline calibration task **calnicb,** and is the same as the association table _asn, with the addition of new columns which report the offsets between different images of the mosaic or chop pattern as calculated by **calnicb**, and the background levels computed for each image. This file is only produced for an associated set of observations.
- *Trailer File* (_trl):
  This FITS ASCII table contains a log of the pipeline calibration processing that was performed on individual datasets and mosaic products.
- *Processing Data Quality File* (_pdq):
  This FITS ASCII table provides quality information on the observation, mostly on pointing and guide star lock. Possible problems encountered, e.g., a loss of guide star lock or a guide star acquisition failure, are reported here.

2.1.2 Science Data Files

The *_raw.fits, *_cal.fits, *_ima.fits, and *_mos.fits files are all defined as *science data files*, as they contain the images of interest for scientific analysis.

File Contents and Organization

The data for an individual NICMOS science readout consist of five arrays, each stored as a separate image extension in the FITS file. The five data arrays represent:

- The science (SCI) image from the detector.
- An error (ERR) array containing an estimate of the statistical uncertainties (in units of 1σ) of the science data.
- An array of bit-encoded data quality (DQ) flags representing known status or problem conditions of the science data.

- An array containing the number of data samples (SAMP) that were used to compute each science image pixel value.
- An array containing the effective integration time (TIME) for each science image pixel.

A grouping of the five data arrays for one science image is known as an *image set* or *imset*.

A science data file can contain one or more imsets. For example, an individual NICMOS exposure obtained with the ACCUM mode will generate a `*_raw.fits` file with one imset; an individual MULTIACCUM exposure with *n* readouts will generate a `*_raw.fits` file (and, after calibration, an `*_ima.fits` file) containing *n* + 1 imsets, including the zeroth readout. The `*_cal.fits` and `*_mos.fits` files always contain one imset.

The insets of a MULTIACCUM exposure are stored in files from last to first. The result of the longest integration time (the last readout performed on-board) occurs first in the file (first imset), the readout before the last is the second imset, and so on; the zeroth readout is the last imset.

Although the five science, error, data quality, samples, and integration time arrays associated with each imset are stored in a single file, they are kept separate as five individual FITS image *extensions* within the file. The order of the images in the FITS files is listed in table 2.1 and shown graphically in figure 2.1 and figure 2.2. The examples given in table 2.1 and figure 2.2 refer to a MULTIACCUM image (multiple imsets), while figure 2.1 refers to ACCUM and BRIGHTOBJ images (one imset, namely extensions 1 through 5). Each extension comes with its own header, and each FITS file contains, in addition, a *global* or *primary header* (primary header-data unit or HDU).

The only contents of the primary HDU are header keywords. There are no image data in the primary HDU. The keywords in the primary header are termed *global keywords* because they apply to the data in all of the file extensions. The organization and location of header keywords is explained in detail later in the chapter.

Table 2.1: NICMOS Science Data File Contents

Header-Data Unit	Extension Name	imset	Contents	Data Type
Primary (Extension 0)	(N/A)	(N/A)	Global keywords; no data.	(N/A)
Extension 1	SCI	1	Science image	raw: 16-bit int; calibrated: float
Extension 2	ERR	1	Error (sigma) image	float
Extension 3	DQ	1	Data Quality image	16-bit int
Extension 4	SAMP	1	Number of Samples image	16-bit int
Extension 5	TIME	1	Integration Time image	float
Extension 6	SCI	2	Science image	raw: 16-bit int; calibrated: float
Extension 7	ERR	2	Error (sigma) image	float
Extension 8	DQ	2	Data Quality image	16-bit int
Extension 9	SAMP	2	Number of Samples image	16-bit int
Extension 10	TIME	2	Integration Time image	float

Figure 2.1: Data Format for ACCUM, BRIGHTOBJ and ACQ Modes

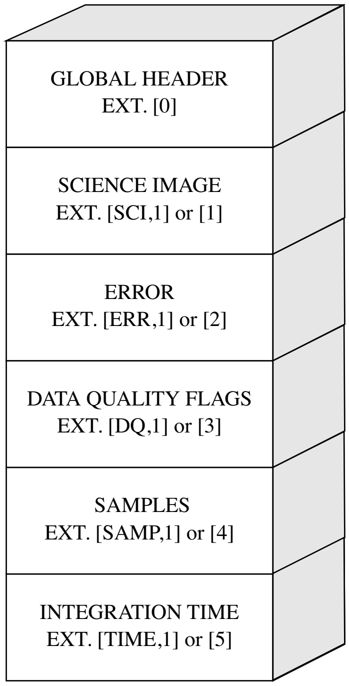
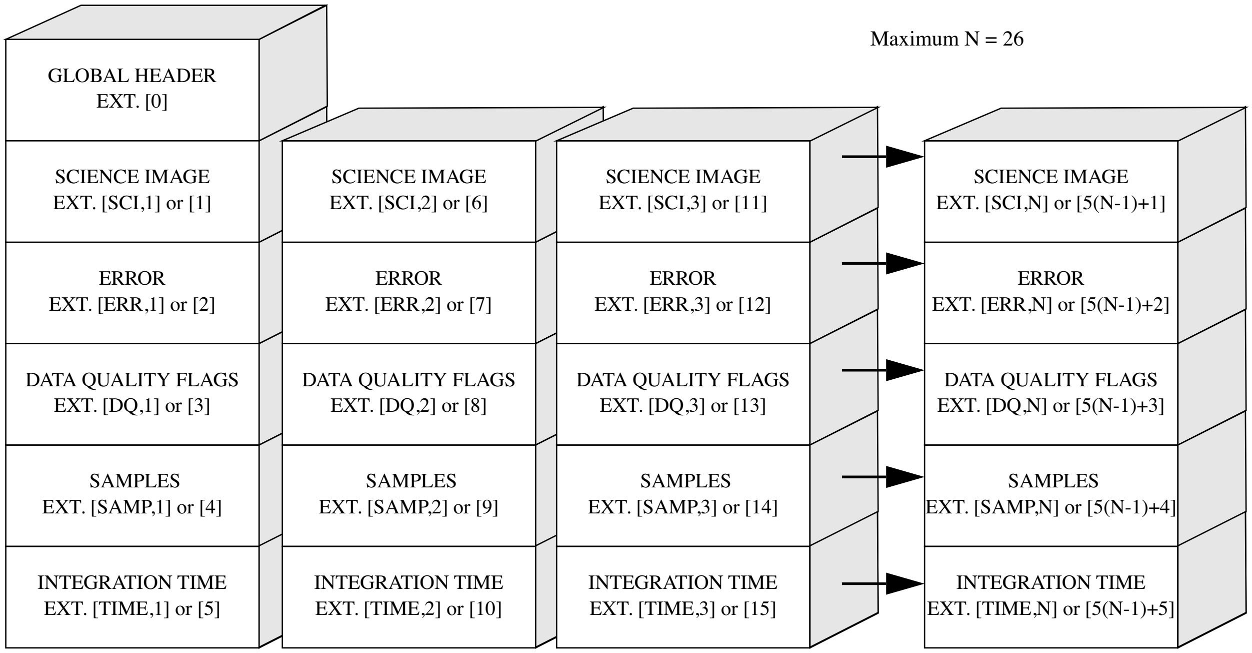


Figure 2.2: MULTIACCUM Mode Data Format



The following sections explain the contents and origin of each of the five image arrays in each imset in more detail.

Science Image

This image contains the data from the detector readout. In ACCUM and BRIGHTOBJ modes the image received from the instrument is the result of subtracting the initial from the final readouts of the exposure. In MULTIACCUM mode the images received are the raw (unsubtracted) data corresponding to each detector readout. In raw datasets the science array is an integer (16-bit) image in units of DNs (counts). In calibrated datasets it is a floating-point image in units of DNs per second (count rates).

Error Image

The error image is a floating-point array containing an estimate of the statistical uncertainty associated with each corresponding science image pixel. This image is computed in the ground calibration pipeline task **calnica** as a combination of detector read noise and Poisson noise in the accumulated science image counts (see chapter 3) and is expressed in terms of 1σ uncertainties. For calibrated MULTIACCUM images (i.e., `*_cal.fits` files), the values of the error array are computed uncertainties in the count rates derived from the linear fit to counts vs. exposure time from the intermediate readouts.

Data Quality Image

This integer (unsigned 16-bit) array contains bit-encoded data quality flags indicating various status and problem conditions associated with corresponding pixels in the science image. Because the flag values are bit-encoded, a total of 16 simultaneous conditions can be associated with each pixel. Table 2.2 lists the flag values and their meanings.

Table 2.2: NICMOS Data Flag Values

Flag Value	Bit Setting[1]	Flag Meaning
0	0000 0000 0000 0000	No known problems
1	0000 0000 0000 000**1**	Reed-Solomon decoding error in telemetry
2	0000 0000 0000 00**1**0	Poor or uncertain Linearity correction
4	0000 0000 0000 0**1**00	Poor or uncertain Dark correction
8	0000 0000 0000 **1**000	Poor or uncertain Flat Field correction
16	0000 0000 000**1** 0000	Pixel affected by “grot” on the detector
32	0000 0000 00**1**0 0000	Defective (hot or cold) pixel
64	0000 0000 0**1**00 0000	Saturated pixel
128	0000 0000 **1**000 0000	Missing data in telemetry
256	0000 000**1** 0000 0000	Bad pixel determined by calibration
512	0000 00**1**0 0000 0000	Pixel contains Cosmic Ray
1024	0000 0**1**00 0000 0000	Pixel contains source (see section 3.4)
2048	0000 **1**000 0000 0000	Pixel has signal in 0th read (see section 3.3)
4096	000**1** 0000 0000 0000	User flag
8192	00**1**0 0000 0000 0000	User flag
16384	0**1**00 0000 0000 0000	Reserved

[1] Most significant bit is at left.

Number of Samples Image

The SAMP image is an integer (16-bit) array containing the total number of data samples that were used to compute the corresponding pixel values in the science image. For ACCUM and BRIGHTOBJ modes, the number of samples contributing to each pixel always has a value of 1 in the raw data file. For MULTIACCUM mode the sample values in the raw and intermediate data files are set to the number of readouts that contributed to the corresponding science image.

Because the number of samples in the raw images for MULTIACCUM, ACCUM and BRIGHTOBJ modes is the same value for all pixels of an imset, the image array is usually not created (to save on data volume), and the value of the sample is stored in the header keyword PIXVALUE in the SAMP image extension (see table 2.4 below).

In MULTIACCUM calibrated data files (`*_cal.fits`) the SAMP array contains the total number of valid samples used to compute the final science image pixel value, obtained by combining the data from all the readouts and rejecting cosmic ray hits and saturated pixels. In this case the sample array may have different values at different pixel locations (less than or equal to the total number of samples in the MULTIACCUM sequence), depending on how many valid samples there are at each location.

In the mosaic images (`*_mos.fits`), the data in the SAMP array indicate the number of samples that were used from overlapping images to compute the final science image pixel value.

Integration Time Image

The TIME image is a floating-point array containing the effective integration time associated with each corresponding science image pixel value. These data are always computed in the ground calibration pipeline for recording in the raw data file.

For ACCUM and BRIGHTOBJ mode observations, each pixel has the same time value. For MULTIACCUM observations each pixel for a given readout has the same time value in the raw and intermediate data. In these cases, to save on data volume, the image array is not created and the value of the time is stored in the header keyword PIXVALUE in the TIME image extension (see table 2.4).

In MULTIACCUM calibrated data files (`*_cal.fits`) the TIME array contains the combined exposure time of all the readouts that were used to compute the final science image pixel value, after rejection of cosmic ray and saturated pixels from the intermediate data. As in the case of the SAMP array, the TIME array can have different values at different pixel locations, depending on how many valid samples compose the final science image in each pixel.

In mosaic images (*_mos.fits), the TIME array values indicate the total effective exposure time for all the data from overlapping images that were used to compute the final science image pixel values.

2.1.3 Auxiliary Data Files

The *_spt.fits, *_trl.fits, *_pdq.fits, *_asn.fits, and the *_asc.fits files are termed *auxiliary data files*. They contain supporting information on the observation, such as spacecraft telemetry and engineering data, assessment of the quality of the observation, calibration information, and information on the associations present in the observations.

Association Tables

The *association tables*, *_asn.fits and *_asc.fits, are FITS binary tables which are created when a particular observation generates an association of datasets (see the discussion of "Associations" in appendix B). In particular, the *_asn.fits table is generated by OPUS[1], and contains the list of datasets which make up the association (e.g., from a dither or chop pattern). The *_asn.fits tables are the inputs to the pipeline **calnicb**, which creates the mosaiced or background subtracted images (*_mos.fits files) from the input datasets. All the datasets must have been processed through the basic pipeline data reduction (**calnica**) before being processed through **calnicb**. In addition to the output science image(s), **calnicb** produces another association table (*_asc.fits), which has the same content as the *_asn.fits table, along with additional information on the offsets used by the pipeline for reconstructing the science image and background values computed for each image. For mosaics (dither patterns), there is only one final image produced, with file name *0_mos.fits. For chop patterns, in addition to the background-subtracted image of the target (*0_mos.fits), an image for each background position is produced; the file names of these background images are *1_mos.fits, *2_mos.fits,..., *8_mos.fits (a maximum of eight independent background positions is obtainable with the NICMOS patterns; see the *NICMOS Instrument Handbook* for details).

Support File

The support files, *_spt.fits, contain information about the observation and engineering data from the instrument and spacecraft that was recorded at the time of the observation. A support file can have multiple FITS image extensions within the same file; in the case of a MULTIACCUM observation there will be one extension for each readout (i.e., each imset) in the science data file. Each extension in the support file

[1] Observation Support & Post Observation Data Processing Unified Systems

holds an integer (16-bit) image containing the data which populates the `*_spt.fits` header keyword values. One example of useful information from the support files is detector temperature information, which can be important when matching DARK reference files to the images (see, e.g., section 4.1.3).

Trailer File

The trailer files, `*_trl.fits`, contain information on the calibration steps executed by the pipelines and diagnostics issued during the calibration. There is one `*_trl.fits` file produced for each dataset, and, in the case of associations, one `*_trl` file for each NICMOS product (i.e., each `*_mos.fits` file).

Processing Data Quality File

The processing data quality files, `*_pdq.fits`, contain general information summarizing the observation, a data quality assessment section, and a summary on the pointing and guide star lock. They state whether problems were encountered during the observations, and, in case they were, describe the nature of the problem. There is one `*_pdq.fits` file produced for each dataset, and, in case of associations, one `*_pdq` file for each NICMOS product (i.e., each `*_mos.fits` file).

2.2 Header Keywords

Both the primary header and the headers of each image extension in a science data file contain *keywords*. The keywords store a wide range of information about the observations themselves (e.g., observing mode, integration time, filters or grisms used), the processing of the data by the OPUS pipeline (e.g., calibration steps performed and reference files used), and the properties of the data themselves (e.g., number of image extensions, dimensions and data type of each image, coordinate system information, flux units, and image statistics).

The primary header carries global keywords which are applicable to all extensions. The extension headers carry extension-specific keywords, which contain information relevant to the image in a particular extension. For example, observation parameters, calibration switches and reference file names are contained in the primary header. Exposure time and World Coordinate System information, on the other hand, are contained in the header of each image extension because this information could vary from one set of extensions to another.

Table 2.3 below lists most of the keywords in the primary header of the *science data files*. The entries in this table are appropriate for data retrieved now from the HST Archive via on-the-fly reprocessing (OTFR - see chapter 3). Some header keywords in older NICMOS data may be different. In particular, the observing pattern keywords have been changed for Cycle 11, and the dictionary below describes the new keyword format. In addition, the tables list the SAA keywords that will be used with the new post-SAA darks to be taken in Cycle 11. For data taken in Cycle 7 and 7N, these keywords will not be populated. The STScI Data Processing Team maintains a complete HST FITS header keyword dictionary with descriptions of the keywords used by NICMOS and other HST instruments.

Table 2.3: Science Data File Primary Header Keywords

Keyword Name	Meaning
	Image Keywords
NEXTEND	Number of extensions in the file (up to 130 for MULTIACCUM).
DATE	Date on which the file was generated
FILENAME	Name of the file
FILETYPE	Type of data: SCI - Science Data File; SPT - Support File; ASN_TABLE - Association Table
TELESCOP	HST
INSTRUME	Instrument used (NICMOS)
EQUINOX	Equinox of the celestial coordinate system (J2000.0 for HST observations)
	Data Description Keywords
ROOTNAME	Rootname (IPPPSSOOT) of the dataset
IMAGETYP	Image type: EXT = external image; FLAT = flatfield image; DARK = dark image
PRIMESI	Primary Instrument used for the observation
	Target Information
TARGNAME	Proposer’s target name
RA_TARG	Right Ascension of the target (degrees, J2000)
DEC_TARG	Declination of the target (degrees, J2000)
	Proposal Information
PROPOSID	Proposal’s identification number
LINENUM	Exposure’s logsheet line number, from the Phase 2 proposal
PR_INV_L	Principle investigator last name
PR_INV_F	Principle investigator first name
PR_INV_M	Principle investigator middle initial
	Exposure Information
ORIENTAT	Position angle of the image y axis (degrees East of North)
SUNANGLE	Angle between sun and V1 axis
MOONANGL	Angle between moon and V1 axis
FGSLOCK	Commanded FGS lock (FINE, COARSE, GYROS, UNKNOWN)
DATE-OBS	UT date of start of the observation (dd/mm/yy or yyyy-mm-dd)
TIME-OBS	UT time of start of the observation (hh:mm:ss)
EXPSTART	Exposure start time (Modified Julian Date)
EXPEND	Exposure end time (Modified Julian Date)
EXPTIME	Total integration time (sec)
EXPEND	Exposure interruption indicator (e.g., NORMAL)
	Instrument Configuration Information
CAMERA	NICMOS camera used in the observation (1, 2, or 3)

Keyword Name	Meaning
PRIMECAM	NICMOS Prime Camera during the observation (for internal parallels)
FOCUS	In-focus camera for this observation
APERTURE	Aperture used in the observation (NICi, NICi-FIX, NIC2-CORON)
OBSMODE	Observing mode (MULTIACCUM, ACCUM,...)
FILTER	Filter or grism used
NUMITER	Number of iterations of the exposure
NREAD	Number of ACCUM initial and final readouts
NSAMP	Number of MULTIACCUM or RAMP samples
SAMP_SEQ	MULTIACCUM exposure time sequence name
FOMXPOS	X offset of the Camera FOV using NICMOS FOM (arcsec)
FOMYPOS	Y offset of the Camera FOV using NICMOS FOM (arcsec)
NFXTILTP	FOM X-tilt position (arcsec)
NFYTILTP	FOM Y-tilt position (arcsec)
NPXTILTP	PAM X-tilt position (arcsec)
NPYTILTP	PAM Y-tilt position (arcsec)
NPFOCUSP	PAM focus position (mm)
TIMEPATT	Timing pattern identifier for readout
READOUT	Detector array readout rate (FAST, SLOW)
SAMPZERO	Sample time of MULTIACCUM zeroth read (sec)
HCLKRATE	Horizontal clock rate (microseconds)
VIDEO_BW	Readout video bandwidth (kHz)
ADCZERO	Analog-digital coversion zero level (DN, usually some large negative number)
ADCGAIN	Analog-digital conversion gain (electrons/DN)
	Photometry Keywords
PHOTMODE	Combination of <INSTRUMENT>+<CAMERA>+<FILTER>
PHOTFLAM	Inverse sensitivity (erg/cm^2/angstrom/DN)
PHOTFNU	Inverse sensitivity (Jy*sec/DN)
PHOTZPT	ST magnitude system zero point
PHOTPLAM	Pivot wavelength of the photmode (Angstrom)
PHOTBW	Root Mean Square bandwidth of the photmode (Angstrom)
	SAA keywords (for use with post-SAA darks in Cycle 11+)
SAA_EXIT	Time of last exit from SAA contour level 23
SAA_TIME	Seconds since last exit from SAA contour 23
SAA_DARK	Association name for post-SAA dark exposures
SAACRMAP	SAA cosmic ray map file
	Calnica Calibration Reference Files (inputs to calnica)
MASKFILE	Static data quality file
NOISFILE	Detector read noise file
NLINFILE	Detector non-linearity file
DARKFILE	Dark current file
FLATFILE	Flat field file
PHOTTAB	Photometric calibration table
BACKTAB	Background model parameters table
	Calnica Calibration Reference File Pedigree (outputs from calnica)

Keyword Name	Meaning
MASKPDGR	Static data quality file pedigree (values: GROUND dd/mm/yyyy - for reference files originated from Thermal Vacuum data; INFLIGHT dd/mm/yyyy - for reference files originated from on-orbit calibration observations)
NOISPDGR	Detector read noise file pedigree (values: GROUND dd/mm/yyyy; INFLIGHT dd/mm/yyyy)
NLINPDGR	Detector non-linearity file pedigree (values: GROUND dd/mm/yyyy; INFLIGHT dd/mm/yyyy)
DARKPDGR	Dark current file pedigree (values: GROUND dd/mm/yyyy; INFLIGHT dd/mm/yyyy; and MODEL dd/mm/yyyy for the synthetic darks, see chapter 3)
FLATPDGR	Flat field file pedigree (values: GROUND dd/mm/yyyy; INFLIGHT dd/mm/yyyy)
PHOTPDGR	Photometric calibration table pedigree (values: GROUND dd/mm/yyyy; INFLIGHT dd/mm/yyyy)
BACKPDGR	Background model parameters table pedigree (values: GROUND dd/mm/yyyy; INFLIGHT dd/mm/yyyy)
	Calnica Calibration Switches (allowed values: PERFORM, OMIT)
BIASCORR	Correct wrapped pixel values
ZSIGCORR	MULTIACCUM zero read signal correction
ZOFFCORR	Subtract MULTIACCUM zero read
MASKCORR	Data quality initialization (DQ array)
NOISCALC	Calculate statistical errors (ERR array)
NLINCORR	Correct for detectors non-linearities
DARKCORR	Dark correction
BARSCORR	Bars correction
FLATCORR	Flat-field correction
UNITCORR	Convert to count rate
PHOTCALC	Populate photometry keywords
CRIDCALC	Identify cosmic ray hits (update of DQ arrays in `*_ima.fits` output of **calnica** for MULTIACCUM)
BACKCALC	Calculate background estimates
WARNCALC	Generate user warnings
	Calnica Calibration Indicators (output from calnica; values: PERFORMED, SKIPPED, OMITTED)
BIASDONE	Correct wrapped pixel values
ZSIGDONE	MULTIACCUM zero read signal correction
ZOFFDONE	Subtract MULTIACCUM zero read
MASKDONE	Data quality initialization (DQ array)
NOISDONE	Calculate statistical errors (ERR array)
NLINDONE	Correct for detectors non-linearities
DARKDONE	Dark correction
BARSDONE	Bars correction
FLATDONE	Flat-field correction
UNITDONE	Convert to count rate
PHOTDONE	Populate photometry keywords
CRIDDONE	Identify cosmic ray hits (update of DQ arrays in `*_ima.fits` output of **calnica** for MULTIACCUM)
BACKDONE	Calculate background estimates
WARNDONE	Generate user warnings

Keyword Name	Meaning
	Calibration Status
CALSTAGE	State of the calibration (values: CALNICA, CALNICB, UNCALIBRATED)
CAL_VER	Version number of the CALNICA (CALNICB) code for `_cal.fits` (`_mos.fits`) files
PROCTIME	Pipeline processing time (MJD)
	Calnicb Calibration Information
ILLMCORR	Subtraction of background illumination pattern reference image (input values: PERFORM, OMIT)
ILLMDONE	Subtraction of background illumination pattern reference image (output values: PERFOMED, SKIPPED, OMITTED)
ILLMFILE	Background illumination pattern reference image filename
ILLMPDGR	Background illumination pattern file pedigree
MEAN_BKG	Mean background level (DN/sec), computed by **calnicb**
	Pattern Keywords
PATTERN1	Primary pattern type
P1_SHAPE	Primary pattern shape
P1_PURPS	Primary pattern purpose
P1_NPTS	Number of points in primary pattern
P1_PSPACE	Point spacing for primary pattern (arc-sec)
P1_LSPACE	Line spacing for primary pattern (arc-sec)
P1_ANGLE	Angle between sides of parallelogram pattern (deg)
P1_FRAME	Coordinate frame of primary pattern (POS-TARG or CELESTIAL)
P1_ORIENT	Orientation of pattern to coordinate frame (deg)
P1_CENTER	Center pattern relative to pointing (yes/no)
BKG_OFF	Pattern offset method (SAM or FOM)
PATTSTEP	Position number of this point in the pattern
PATTERN	Pattern type
	Association Keywords
ASN_ID	Association rootname
ASN_TAB	Name of the association table
ASN_MTYP	Role of the dataset in the association (e.g., target, first background, second background, etc.; allowed values: EXP_TARG, EXP_BCKn, PROD_TARG, PROD_BCKn, where EXP=input exposure, PROD=output product, TARG=target, BCK=background, n=1-8)

In the SCI image extensions, additional keywords describing the data quality are present. They give the number of pixels which have a flag different from zero in the DQ extension, and a suite of statistical information (mean, standard deviation, minimum and maximum of good pixels in the entire detector and per quadrant) on the image. Table 2.4 lists some of the relevant keywords that are specific to image extensions; they appear in the extension headers, but not in the primary header.

Table 2.4: Image Extension Header Keywords in Science Data Files

Keyword Name	Meaning
	Data Description Keywords
EXTNAME	Name of the extension in an imset of the data file (SCI, ERR, DQ, SAMP, TIME)
EXTVER	Extension version; integer number to uniquely identify an IMSET in a science data file. A MULTIACCUM file can contain up to 26 IMSETs, i.e. up to EXTVER=26.
INHERIT	Switch to allow the image extension header to inherit the primary header keywords. Allowed values: T=TRUE, F=FALSE.
DATAMIN	Minimum pixel value
DATAMAX	Maximum pixel value
BUNIT	Brightness units; allowed values: COUNTS, COUNTS/S.
PIXVALUE	When ALL pixels in an image extension have the same value (e.g., the SAMP and TIME arrays in the *_ima.fits file from a MULTIACCUM exposure or the ERR, DQ, SAMP and TIME arrays in the *_raw.fits files from a MULTIACCUM, ACCUM or BRIGHTOBJ exposure) the pixel array of that extension is not generated, and the PIXVALUE keyword is instead populated with the common value of the pixels to save space.
	World Coordinate System of Image
CRPIX1	x-coordinate of image’s reference pixel
CRPIX2	y-coordinate of image’s reference pixel
CRVAL1	RA of reference pixel (degrees)
CRVAL2	DEC of reference pixel (degrees)
CD1_1	Partial derivative of RA with respect to x
CD1_2	Partial derivative of RA with respect to y
CD2_1	Partial derivative of Dec with respect to x
CD2_2	Partial derivative of Dec with respect to y
	Readout Parameters
SAMPNUM	Sample number of the MULTIACCUM sequence
SAMPTIME	Total integration time (sec)
DELTATIM	Integration time of the sample (sec)
ROUTTIME	UT time of array readout (MJD)

Keywords in the support file, `*_spt.fits`, report information from the ephemeris and engineering data on the status of the telescope and of the instrument during the observations. Information common to all readouts of a MULTIACCUM sequence are stored in the primary header of the support file, while information which may change from readout to readout (e.g., detector temperature) is stored in the headers for each corresponding extension. Table 2.5 describes some of the relevant primary header keywords from the *_spt.fits file, and table 2.6 gives some useful keywords from the extension headers.

Table 2.5: NICMOS Primary Header Keywords in the Support Files

Keyword Name	Meaning
PA_V3	Position angle of the V3 axis of HST (degrees)
PA_V3	RA of the V1 axis of HST (degrees in J2000)
RA_V1	DEC of the V1 axis of HST (degrees in J2000)
DEC_V1	RA of the Sun (degrees in J2000)
RA_SUN	DEC of the Sun (degrees in J2000)
DEC_SUN	RA of the Moon (degrees in J2000)
RA_MOON	DEC of the Moon (degrees in J2000)

Table 2.6: Some Useful NICMOS Extension Header Keywords in the Support Files

Keyword Name	Meaning
NDWTMP11	Camera 1 and 2 Mounting Cup temp (deg K)
NDWTMP13	Camera 3 Mounting Cup temp (deg K)

2.3 Working with NICMOS Files

The quickest way to learn how each observation was performed is to use the **iminfo** task in the STSDAS **toolbox.headers** package to look at the headers of the science data. The output from **iminfo** summarizes on one screen the relevant information about an observation (table 2.7) and the instrument configuration during the observations (table 2.8), by reading and reporting the values of various keywords.

Table 2.7: Observation Information in **iminfo** Listing

Field Descriptor	Header Keyword Source
Rootname	ROOTNAME
Instrument	INSTRUME
Target Name	TARGNAME
Program	ROOTNAME (positions 2–4)
Observation set	ROOTNAME (positions 5–6)
Observation	ROOTNAME (positions 7–8)
File Type	FILETYPE
Obs Date	DATE-OBS or FPKTTIME
Proposal ID	PROPOSID
Exposure ID	PEP_EXPO
Right Ascension	CRVAL1
Declination	CRVAL2
Equinox	EQUINOX

Table 2.8: NICMOS-Specific Information in **iminfo** Listing

Field Descriptor	Header Keyword Source
Image type	IMAGETYP
Number of extensions	NEXTEND
Camera number	CAMERA
Aperture	APERTURE
Filter name	FILTER
Observation Mode	OBSMODE
Number of initial/final reads	NREAD (ACCUM)
Number of intermediate samples	NSAMP (MULTIACCUM)
MULTIACCUM sequence	SAMP_SEQ
Exposure time (sec)	EXPTIME
Readout speed	READOUT
Association ID	ASN_ID
Number of Iterations	NUMITER
Calibration steps done	Switches whose values are set to “PERFORMED”. Switches are: ZSIGDONE, ZOFFDONE, MASKDONE, BIASDONE, NOISDONE, DARKDONE, NLINDONE, BARSDONE, FLATDONE, UNITDONE, PHOTDONE, CRIDDONE, BACKDONE, WARNDONE

The entire suite of keywords from any header can be listed with the IRAF task **imheader**. Given that NICMOS data files contain multiple extensions, the number of the desired extension must always be specified. For example, to list the primary header content of a calibrated image, you type

```
cl> imheader n0g70106t_cal.fits[0] long+ | page
```

where [0] identifies the primary header. To list the header of the *second* science image in a MULTIACCUM sequence (the sixth extension):

```
cl> imheader n0g70106t_cal.fits[6] long+ | page
```

Chapter 2 of the HST Introduction describes in detail how to work with FITS file extensions. Here we will recap the essentials. In order to simplify access to NICMOS FITS image extensions, each extension header contains the two keywords EXTNAME (extension name) and EXTVER (extension version number). The EXTNAME keyword identifies the nature of the extension (SCI, ERR, DQ, SAMP, TIME, see table 2.7). The EXTVER keyword contains an integer value which is used to uniquely identify a particular imset (quintuple of image extensions). For example, the five image extensions (single imset) contained in the science data file for an ACCUM or BRIGHTOBJ observation will all usually be assigned an EXTVER value of 1 because there will only be one set of extensions in the file. In a MULTIACCUM science data file, each set of extensions associated with a given readout will have a unique EXTVER value, running from 1 up to the total number of readouts in that particular file.

To list the header of the *second* science image in a MULTIACCUM sequence, in place of the command line above, one could instead type:

```
cl> imheader n0g70106t_cal.fits[sci,2] long+ | page
```

In general, to access a particular image extension, append the name and version number of the desired extension in square brackets to the end of the file name. The EXTNAME value is specified first, then the EXTVER value, separated by a comma. Indeed, the use of the keywords EXTNAME and EXTVER is not limited to the task **imheader,** but can be used in all IRAF tasks.

The primary header data unit in a NICMOS FITS file does not contain the EXTNAME or EXTVER keywords. The absolute extension number 0 (zero) refers to the primary header.

If a calibration keyword needs to be changed, the IRAF/STSDAS **chcalpar** task can be used. For instance, to modify the flatfield calibration switch from PERFORM to OMIT in a given data file, the following command can be given:

```
cl> chcalpar n0g70106t_raw.fits
```

The parameter set (or "pset") list appropriate for the image will appear, and the calibration keyword can be modified. The operation performed with **chcalpar** can be equivalently performed (although in a more cumbersome way) with the general IRAF task **hedit**; in this case, the extension [0] of the primary header *must* be explicitly specified:

```
cl> hedit n0g70106t_raw.fits[0] flatcorr OMIT
```

Do not try to edit a keyword in an extension header unless you are certain that the keyword does not reside in the primary header.

Image sections can be specified in the case of NICMOS data with the same syntax as all IRAF images. For example, to specify a pixel range from 101 to 200 in the *x* direction and all pixels (denoted by an asterisk) in the *y* direction from the second error image in a file, the complete file name specification would be:

```
n0g70106t_cal.fits[err,2][101:200,*].
```

If you use both extension and image section syntax together, the extension name or number must come first, enclosed in one set of brackets, and the image section specification in a second set of brackets.

2.4 From the Phase II Proposal to Your Data

The connection between the Exposure Logsheet that each observer fills out during the Phase II proposal process and the datasets and associations that the observer receives once the observations are executed can be better understood through some examples.

The first example shows an exposure logsheet entry that will generate only one dataset:

```
Exposure_Number: 1
Target_Name: HDF
Config: NIC2
Opmode: MULTIACCUM
Aperture: NIC2
Sp_Element: F160W
Optional_Parameters: SAMP-SEQ=STEP256,NSAMP=12
NUMBER_of_Iterations: 1
Time_Per_Exposure: DEF
Special_Requirements: POS TARG 0.5, 0.5
```

The science data file in the dataset will contain 13 imsets (corresponding to the MULTIACCUM NSAMP=12 parameter plus the 0'th read), and some of the header keywords will be filled with the relevant information from the target and exposure logsheets of the Phase II (e.g., the keywords TARGNAME, RA_TARG, DEC_TARG,....).

The next example shows an exposure logsheet entry that will generate both multiple datasets and an association:

```
Exposure_Number: 1
Target_Name: HDF
Config: NIC2
Opmode: MULTIACCUM
Aperture: NIC2
Sp_Element: F160W
Optional_Parameters:PATTERN=SPIRAL-DITH-CHOP,NUM-POS=8,
DITH-SIZE=1.5,CHOP-SIZE=25.0,SAMP-SEQ=STEP256,NSAMP=12
Number_of_Iterations: 1
Time_Per_Exposure: DEF
Special_Requirements:
```

In this observation, eight datasets (one for each position of the pattern) and one association will be created. The pipeline products will include the eight reduced datasets, one mosaic of the background-subtracted target, and one mosaic of the background.

An association will be generated also in the case below:

```
Exposure_Number: 1
Target_Name: HDF
Config: NIC2
Opmode: MULTIACCUM
Aperture: NIC2
Sp_Element: F160W
Optional_Parameters: SAMP-SEQ=STEP256,NSAMP=12
NUMBER_of_Iterations: 3
Time_Per_Exposure: DEF
Special_Requirements:
```

The number of iterations is 3, implying that three datasets will be generated from this exposure logsheet, plus an association table containing the three datasets. The collection of multiple iterations into an association is a new feature introduced by the NICMOS and STIS pipelines. In our specific example, the co-added image from the three iterations will be one of the products of the pipeline.

2.5 Paper Products

After the data from an observation have been received and processed through the STScI pipeline, PDF files (*paper products*) are automatically generated which summarize the data obtained. The paper products may also be generated by the observer using the **stsdas.hst_calib.paperprod** task **pp_dads** to provide a first look at the observations and their quality. Here we briefly describe the NICMOS paper products.

As of 1 August 1999, the Archive is no longer printing paper copies of the "paper products" for new observations. The PDF files must now be retrieved via the WWW from http://archive.stsci.edu/hst/pdf_search.html.

Paper products typically summarize the set of exposures that constitute a visit in the Phase II proposal. The set of exposures can be either individual datasets or associations. Paper products are produced by accessing the appropriate keywords in the dataset headers or in the association tables.

A given page of the NICMOS paper products falls into one of two categories: visit-level page or exposure-level page. The content of the pages is as follows:

Visit-Level Pages

- ***Cover Page***: contains the proposal ID, the visit number, the PI's last name, and the proposal title.
- ***Explanatory Notes***: a set of notes explaining the information contained in the paper products.
- ***Target List***: a table listing the targets of the observations being summarized.
- ***Observation Summary***: a table summarizing the proposal information for each exposure in the present set, including processing and data quality flags.
- ***Optional Parameters***: a table listing the optional parameters, other than the pattern related parameters, used in the observations.
- ***Observing Pattern Strategy***: a table listing the observing pattern used for each exposure in the set.

Exposure-level Pages

- ***Final Calibrated Image***: a grey-scale plot of the calibrated science image (a mosaic if the observation was dithered).
- ***Observation Parameters***: several useful parameters are listed on the right hand side of a subset of the paper product pages. Observation root name, the date and time of the observation, the target name and position, the instrument configuration, FOM offset, and pattern, dither and chop information is given.
- ***Spacecraft Performance:*** known problems with guide star acquisition, spacecraft guidance, recenterings, and telemetry drop-outs are listed, as are detected problems with the instrument's operation (e.g., Take Data flag NOT on throughout observation).
- ***Pipeline Processing Summary:*** any problems encountered in the routine pipeline processing of the data are listed.
- ***Calibration Data Quality Summary:*** possible problems with the calibration reference files is summarized; for example, any dummy reference files used in the calibration would be identified.
- ***Thumbnail plots:*** the individual exposures for dithered, chopped, and NUMITER > 1 observations are given.
- ***Observing Pattern:*** a schematic (cartoon) of the observing pattern is shown.
- ***Dither, Chop Mosaics:*** plots of the mosaiced calibrated image (on-target) and average mosaiced background image (off-target) are included.
- ***Data Quality Summary***: a summary of the spacecraft performance, pipeline processing status, and calibration data quality for each exposure.
- ***Calibration Reference File Summary***: a summary of the calibration processing switches and reference files used to process each exposure.

CHAPTER 3:

Calibration

In this chapter. . .

This chapter is designed to help you understand the steps that are performed on your data in the routine pipeline process and to help you decide whether you should recalibrate your data. In this chapter we:

- provide flowcharts and descriptions of the NICMOS pipeline calibration steps, and
- explain how to recalibrate your data using the calibration software in STSDAS.

The next chapter (chapter 4) will discuss a variety of NICMOS data anomalies which may require processing that goes beyond the standard calibration pipeline. You should read both INTRO 3 and chapter 4 carefully before reducing your data in order to be fully aware of the relevant issues and to obtain the best results from your observations.

3.1 Pipeline Processing, OTFR, and the HST Archive

During Cycle 7 and 7N, NICMOS data arriving at STScI followed the standard path for processing and archiving. They passed through the OPUS *pipeline* which processed and calibrated them, and the resulting raw and processed data products were recorded as static entities in the HST

Data Archive. Users interested in reprocessing their NICMOS observations could retrieve the raw data and reference files from the archive and process them at their home institutions using STSDAS software. If you have received NICMOS data as a GO or from the HST Archive before late September 2001, your data was processed in this fashion.

Beginning 26 September 2001, NICMOS data are now retrieved from the HST Archive using "On-The-Fly Reprocessing" (OTFR), a method previously implemented for STIS and WFPC2, and which will be used for all data from future HST instruments as well. OTFR goes back to the original telemetry (the "POD files") to reconstruct FITS files, and then processes these raw FITS data through the most up-to-date version of the OPUS pipeline software, using the latest and best reference files available at the time of retrieval from the Archive. As with all HST instruments, the pipeline software and reference files have evolved and improved with time as our knowledge of the instrument and its calibration have developed. OTFR ensures that users will receive the best standard processing available at the time they request NICMOS data from the Archive, and makes it easy to recalibrate data taken earlier in the mission lifetime.

All of the steps performed by the pipeline are recorded in the trailer file for your dataset (`*_trl.fits`). The main steps performed by the pipeline are:

1. The data are partitioned (separated into individual files, e.g., engineering and science data are separated).
2. The data are edited, if necessary, to insert fill values in place of missing data.
3. The data are evaluated to determine if there are discrepancies between a subset of the planned and executed observational parameters.
4. A list of calibration reference files to be used in the calibration of the data is created based on the executed observational parameters. This step does not generate comments in the NICMOS trailer file.
5. The raw data are converted to a generic (FITS) format and the header keyword values are populated (known as *generic conversion*).
6. The raw data are calibrated using a standard set of NICMOS calibration programs (**calnica** and **calnicb**).

The calibration software used by the pipeline (step 6 above) is exactly the same as that provided within STSDAS (see section 3.2). The calibration files and tables used are taken from the Calibration Data Base System (CDBS) at STScI and are the most up-to-date versions available at the time the data are requested and processed by OTFR. Sometimes, however, improved calibration reference files are created later, and this is one reason why you may wish to reprocess your data (see section 3.5).

3.2 NICMOS Calibration Software

3.2.1 The Calibration Pipeline

The science data that an observer receives are calibrated in the pipeline by at least one, and possibly two, STSDAS calibration routines: **calnica** and **calnicb**. The two routines perform different operations:

1. **calnica:** This routine removes the instrumental signature from the science data. It is the first calibration step, and is applied to *all* NICMOS datasets individually. **Calnica** operates on the raw science data files.
2. **calnicb**: This routine operates on *associations*: it co-adds datasets obtained from multiple iterations of the same exposure, mosaics images obtained from dither patterns, and background-subtracts images obtained from chop patterns. **Calnicb** is applied to the calibrated science data files (output from **calnica**), and requires association tables (`*_asn.fits`) and the `*_spt.fits` files.

Both tasks determine which calibration steps are to be performed by looking at the values of the *calibration switch* keywords in the primary header of the input science data files (see table 2.3). The tasks select the reference files to use in the calibration of the data by retrieving the reference file names from the *reference file* keywords, also located in the primary header of the input data files. The appropriate values of the calibration switches and reference file keywords depend on the instrumental configuration used, the date when the observations were taken, and any special pre-specified constraints. They are set in the headers of the raw data file in the pipeline during the generic conversion process. The *calibration indicators* keywords record which steps have been performed on the data, and get updated after processing. In particular, the indicators for completed steps will have been assigned the value “PERFORMED”, while the indicators for the steps that were not performed will have been set to “OMITTED” or “SKIPPED”. The calibration indicators keywords should be examined in the primary header of the calibrated science data (`*_cal.fits`) to determine what calibration steps were applied to the data.

The **calnica** and **calnicb** tasks are available in STSDAS in the **hst_calib.nicmos** package. By using these tasks, observers can recalibrate data using the same software as the routine calibration pipeline at STScI.

3.2.2 Software for Grism Data Reduction

A unique capability of NICMOS is the grism mode, which permits multi-object, slitless spectroscopy at low resolution. Grism data are processed with separate calibration software, **calnicc**, which performs a series of steps that identify and extract spectra from the images. The inputs to **calnicc** are the calibrated images (`*_cal.fits`) produced by **calnica**.

Calnicc was developed using the Interactive Data Language (**IDL**) software at the Space Telescope European Coordinating Facility (ST-ECF). The manual (written by W. Freudling and R. Thomas) describing the software, its installation, and use can be found at the ST-ECF calnicc web site.

Currently calnicc is not part of the automatic pipeline processing, and users must apply the calibration software to their grism images

Many users may want to start the spectrum extraction processing using **NICMOSlook**, the interactive counterpart to **calnicc**. **NICMOSlook** is written in **IDL**, and is a quick-look spectrum extraction tool for grism spectra. Unlike **calnicc**, **NICMOSlook** requires users to specify parameters interactively (e.g., the best way to find an object, the weights to be given in the spectral extraction). This tool is recommended for first-time users or users with a small number of grism data. Once you are familiar with the extraction process and parameters, use **calnicc**. The **NICMOSlook** user manual (written by W. Freudling, R. Thomas, and L. Yan), the software, and instructions for its installation can be found on the ST-ECF NICMOSLOOK web site.

Because NICMOS grism data processing and extraction are not really "pipeline" procedures, this handbook will defer an overview of grism reduction methodology and the software tools **calnicc** and **NICMOSlook** until chapter 5, where the discussion of NICMOS Data Analysis is presented.

You cannot run either calnicc or NICMOSlook unless you have an IDL licence.

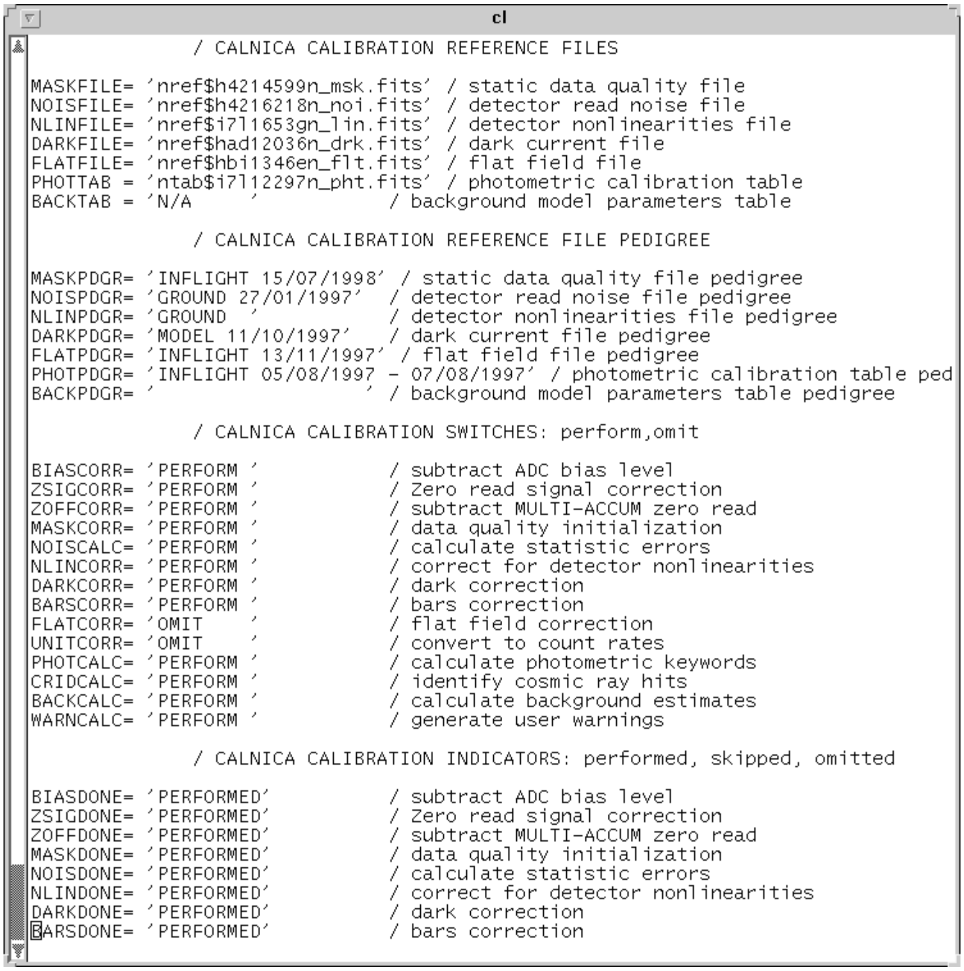
3.3 Basic Data Reduction: calnica

The **calnica** task operates on individual NICMOS datasets and performs the job of removing the instrumental signature from the raw science data. The **calnica** task also tries to identify cosmic ray hits and combines the multiple readouts in MULTIACCUM observations.

The inputs to **calnica** are the raw science (`*_raw.fits`) files. The output of **calnica** is usually a single file containing the *calibrated* science data (`*_cal.fits`). For MULTIACCUM mode datasets there is an additional intermediate output file (`*_ima.fits`) which contains the *calibrated* data from all the intermediate readouts. The `_ima.fits` data are fully calibrated up to, but not including, the cosmic ray rejection. The format of the input and output science data files are identical, so that the output data can be reused as input to **calnica**, if desired. One could, for example, process a science data file through some subset of the normal calibration steps performed by **calnica**, examine or modify the results, and then process the data through **calnica** again, performing other calibration steps or using alternate calibration reference files. One example of such a procedure would be reducing data where there are significant changes in the quadrant bias level from readout to readout in a MULTIACCUM sequence. In section 4.1, we discuss this common NICMOS data anomaly, and in section 4.1.5 we describe one technique for treating it using the **biaseq** task. At present, the use of this routine requires multiple, re-entrant applications of **calnica** in order to partially process the images before and after the use of the **biaseq** task.

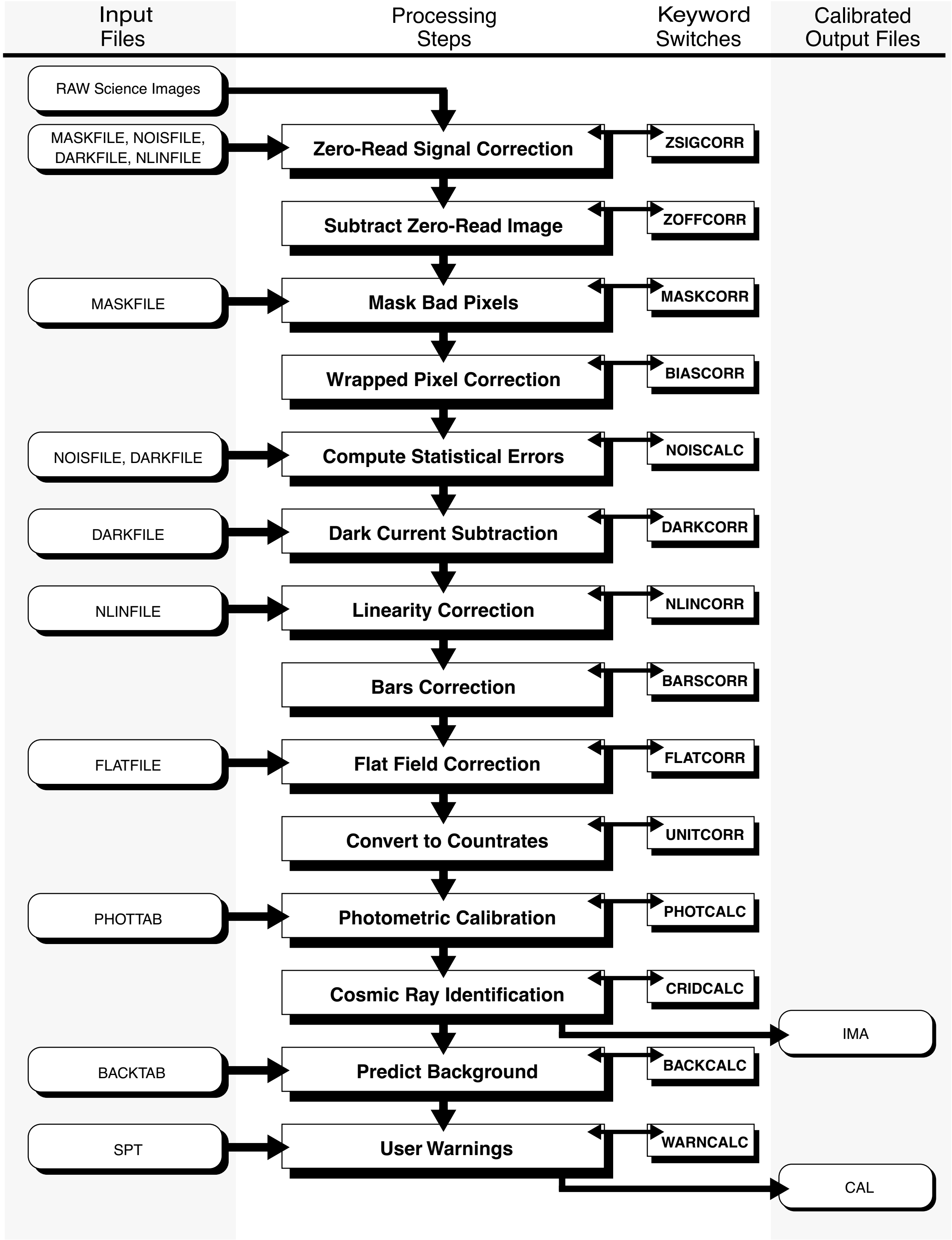
Figure 3.1 shows the portion of a calibrated NICMOS science file header containing the switches and reference file keywords that pertain to the processing performed by **calnica**. The accompanying flow chart (figure 3.2) shows the sequence of **calnica** calibration steps, the input data and reference files and tables, and the output data file. Each calibration step is described in detail in the following sections.

Figure 3.1: Partial NICMOS Header



```
              / CALNICA CALIBRATION REFERENCE FILES

MASKFILE= 'nref$h4214599n_msk.fits' / static data quality file
NOISFILE= 'nref$h4216218n_noi.fits' / detector read noise file
NLINFILE= 'nref$i7l1653gn_lin.fits' / detector nonlinearities file
DARKFILE= 'nref$had12036n_drk.fits' / dark current file
FLATFILE= 'nref$hbi1346en_flt.fits' / flat field file
PHOTTAB = 'ntab$i7l12297n_pht.fits' / photometric calibration table
BACKTAB = 'N/A     '           / background model parameters table

              / CALNICA CALIBRATION REFERENCE FILE PEDIGREE

MASKPDGR= 'INFLIGHT 15/07/1998' / static data quality file pedigree
NOISPDGR= 'GROUND 27/01/1997'  / detector read noise file pedigree
NLINPDGR= 'GROUND  '           / detector nonlinearities file pedigree
DARKPDGR= 'MODEL 11/10/1997'   / dark current file pedigree
FLATPDGR= 'INFLIGHT 13/11/1997' / flat field file pedigree
PHOTPDGR= 'INFLIGHT 05/08/1997 - 07/08/1997' / photometric calibration table ped
BACKPDGR= '                  ' / background model parameters table pedigree

              / CALNICA CALIBRATION SWITCHES: perform,omit

BIASCORR= 'PERFORM '           / subtract ADC bias level
ZSIGCORR= 'PERFORM '           / Zero read signal correction
ZOFFCORR= 'PERFORM '           / subtract MULTI-ACCUM zero read
MASKCORR= 'PERFORM '           / data quality initialization
NOISCALC= 'PERFORM '           / calculate statistic errors
NLINCORR= 'PERFORM '           / correct for detector nonlinearities
DARKCORR= 'PERFORM '           / dark correction
BARSCORR= 'PERFORM '           / bars correction
FLATCORR= 'OMIT    '           / flat field correction
UNITCORR= 'OMIT    '           / convert to count rates
PHOTCALC= 'PERFORM '           / calculate photometric keywords
CRIDCALC= 'PERFORM '           / identify cosmic ray hits
BACKCALC= 'PERFORM '           / calculate background estimates
WARNCALC= 'PERFORM '           / generate user warnings

              / CALNICA CALIBRATION INDICATORS: performed, skipped, omitted

BIASDONE= 'PERFORMED'          / subtract ADC bias level
ZSIGDONE= 'PERFORMED'          / Zero read signal correction
ZOFFDONE= 'PERFORMED'          / subtract MULTI-ACCUM zero read
MASKDONE= 'PERFORMED'          / data quality initialization
NOISDONE= 'PERFORMED'          / calculate statistic errors
NLINDONE= 'PERFORMED'          / correct for detector nonlinearities
DARKDONE= 'PERFORMED'          / dark correction
BARSDONE= 'PERFORMED'          / bars correction
```

Figure 3.2: **Calnica** Processing Flow



ZSIGCORR (Zero-Read Signal Correction)

At the beginning of a NICMOS observation the detector pixels are reset to a bias level and then read out to record that bias level. There is an interval of approximately 0.2 seconds that elapses between the time each pixel is reset and then read. Because NICMOS does not have a shutter, signal from external sources starts to accumulate during that 0.2 second interval. When the initial (or "zeroth") read is later subtracted from subsequent readouts, any signal in the zeroth read will also be subtracted. For very bright sources, the amount of signal in the zeroth read can be large enough to lead to inaccurate linearity corrections, as well as the failure to detect saturation conditions, in the NLINCORR calibration step, because the linearity correction and saturation checking both depend on the absolute signal level accumulated in a pixel.

For MULTIACCUM observations, the ZSIGCORR step is used to estimate the amount of source signal in the zeroth read and to supply corrections to the NLINCORR step for that signal. The ZSIGCORR step estimates the amount of signal in the zeroth read by first measuring the amount of signal that arrived in each pixel between the zeroth and first reads, and then scaling that signal to the effective exposure time of the zeroth read (nominally 0.203 seconds). Pixels that have an estimated zeroth read signal greater than 5 times their ERR value are assumed to contain detectable signal; those below this threshold are ignored. The user may set a different zero-read detection threshold by using the `zsthresh` task parameter for **calnica**. The estimated zeroth read signal is then passed, on a pixel-by-pixel basis, to the NLINCORR step, so that it can account for that signal when applying linearity corrections and saturation checking on the zeroth-read subtracted images with which it works. The ZSIGCORR step also performs saturation checking on the zeroth and first readout images.

Note that this technique will not work well for pixels covered by targets that are so bright that the signal is already beginning to saturate in either the zeroth or first readouts.

Pixels that are determined to have detectable signal in the zeroth read are marked in the DQ images of the output `*_ima.fits` file with a data quality flag value of 2048. The ZSIGCORR routine uses the MASKFILE, NOISFILE, DARKFILE, and NLINFILE reference files.

The ZSIGCORR routine is implemented in calnica versions 3.0 and higher. It was implemented in the standard OPUS calibration pipeline on 11 November 1997, and archived data from before that time does not have the ZSIGCORR step applied. If you are concerned about accurate flux measurements for bright sources in NICMOS observations taken before that time, you may wish to reprocess the data using the latest version of calnica (see section 3.5), or to retrieve the data again from the HST Archive via OTFR, which will automatically apply the ZSIGCORR step.

ZOFFCORR (Subtract Zero-Read Image)

The ZOFFCORR step of **calnica** performs the subtraction of the zeroth read from all readouts in a MULTIACCUM file. This step is performed for data generated by the MULTIACCUM readout mode only. For ACCUM and BRIGHTOBJ readout modes, the subtraction of the zeroth read is performed on-board, because the images returned to the ground are formed by taking the difference of initial and final non-destructive detector readouts.

The pipeline will subtract the zeroth read image from all readouts, including the zeroth read itself. Furthermore, the self-subtracted zeroth-read image will be propagated through the remaining processing steps and included in the output products, so that a complete history of error estimates and data quality (DQ) flags is preserved. After this step is performed, the science data are in the same form as the raw science data from any other observing mode and are processed the same way throughout the remaining steps of **calnica**. No reference files are used by this step.

MASKCORR (Mask Bad Pixels)

Flag values from the static bad pixel mask file are added to the DQ image. This uses the MASKFILE reference file, which contains a flag array for known bad (hot or cold) pixels. There is one MASKFILE for each detector. In general, only truly “defective” pixels are included in the MASKFILE reference images available from the STScI calibration database. There are other bad pixels, however, which you may wish to mask out, such as those pixels affected by “grot” (see section 4.1.5). If you wish, you can create a new MASKFILE which includes additional bad or suspect pixels, and reprocess your data using this new mask.

BIASCORR (Wrapped Pixel Correction)

NICMOS uses 16-bit analog-to-digital converters (ADCs), which convert the analog signal generated by the detectors into signed 16-bit integers. Because the numbers are signed and because the full dynamic range of the converter output is used, raw pixel values obtained from individual detector readouts can range from –32768 to +32767 DN. In practice the detector bias level is set so that a zero signal results in a raw value on the order of –23000 DN. In ACCUM and BRIGHT OBJECT modes, where the difference of initial and final readouts is computed on-board, the subtraction is also performed in 16-bit arithmetic. Therefore, it is possible that the difference between the final and initial pixel values for a bright source could exceed the dynamic range of the calculation, in which case the final pixel value will wrap around the maximum allowed by the 16-bit arithmetic, resulting in a negative DN value. Given the level at which the NICMOS detectors saturate, and the analog-to-digital conversion factor, the maximum "real" pixel value that is expected is on the order of +40000 DN. Such a value will be wrapped to about –26000 DN by the on-board difference calculation.

The BIASCORR step searches for pixel values in the range –23500 to –32768 DN and adds an offset of 65536 DN to these pixel values to reset them to their original real values. The BIASCORR step only affects ACCUM and BRIGHTOBJ mode observations, although it is applied to all NICMOS data sets. For MULTIACCUM data, it should have no effect.

No reference files are used by this step.

NOISCALC (Compute Statistical Errors)

Errors for MULTIACCUM, ACCUM and BRIGHTOBJ modes are computed in the **calnica** pipeline. The NOISCALC step performs the task of computing an estimate of the errors associated with the raw science data based on a noise model for each detector. Currently the noise model is a simple combination of detector read noise and Poisson noise in the signal, such that:

$$\sigma = \frac{\sqrt{\sigma_{rd}^2 + \text{counts} \cdot \text{adcgain}}}{\text{adcgain}}$$

where σ_{rd} is the read noise in units of electrons, *adcgain* is the analog-to-digital conversion gain factor (in electrons per DN) and *counts* is the signal in a pixel in units of DN. Noise is computed in units of electrons, but the result is converted to units of DNs for storage in the error image. The detector read noise is read pixel-by-pixel from the NOISFILE reference image and depends on the read rate of the observation (fast or slow), as well as the number of initial and final reads (NREAD). Separate NOISFILEs are required for each combination of read rate and NREAD. The data quality flags set in the DQ image of the NOISFILE are propagated into the DQ images of all image sets (imsets) being processed.

Because the noise calculation is performed before dark subtraction has taken place, the noiseless electronic signal component known as "shading" (see DARKCORR below) is still present in the data. In **calnica** versions 3.3 and later the NOISCALC step estimates the level of the shading signal in the data by computing column or row statistics in the DARKFILE reference file. The computed shading estimate is subtracted from the signal in the science image when computing Poisson noise on the detected counts. This yields a more accurate noise estimate than what was produced in earlier versions of the pipeline.

Throughout the remaining steps in **calnica**, the error image is processed in lock-step with the science image, getting updated as appropriate. Errors are mostly propagated through combination in quadrature. For MULTIACCUM data sets, the ERR array for the final calibrated image (`*_cal.fits`) is populated by the CRIDCALC step of **calnica** based on the calculated uncertainty of the count rate fit to the MULTIACCUM samples.

In general, the values in the error images should only be regarded as an estimate of the data uncertainties. The precise pixel noise values in NICMOS images are difficult to compute a priori because many factors may contribute, sometimes in unpredictable ways (see, e.g., the discussions of data anomalies such as cosmic ray persistence in chapter 4).

DARKCORR (Dark Current and Bias Shading Subtraction)

Dark images taken with NICMOS contain three distinct, additive signal components: the so-called “shading”, amplifier glow, and the true dark current. The shading is a noiseless signal that appears as gradient across a detector quadrant and is due to the fact that the bias level on the pixels is gradually changing as they are being read out. The amplitude of the shading signal is a function of the time since a pixel was last read out. The amplifier glow is signal produced by a small amount of infrared radiation from the detector readout amplifiers. The amplitude of the amplifier glow is directly proportional to the total number of readouts in an observation. The true detector dark current signal is quite small for the NICMOS arrays and is linearly dependent on the total exposure time of an observation.

Because the shading and amp glow signals depend on factors other than the exposure time of an observation, it is *not* possible to apply a simple scaling of a single dark reference image to match the exposure time of the science data that is being calibrated. Therefore, a library of dark current images is maintained for each of the three cameras, covering all of the predefined MULTIACCUM sample sequences, and a subset of ACCUM exposure times and NREAD values (see the *NICMOS Instrument Handbook*). The reference dark file appropriate for the exposure sequence used in MULTIACCUM, or the exposure time and NREAD values used in ACCUM, is determined by the OPUS generic conversion process when it populates the DARKFILE reference file keyword in the primary header of raw data files. The **calnica** DARKCORR step subtracts the dark reference images, readout-by-readout for MULTIACCUM observations, from the science data. Error estimates of the dark current, stored in the ERR images of the DARKFILE, are propagated in quadrature into the ERR images of all processed science imsets. Data quality (DQ) flags set in the DARKFILE are also propagated into the DQ images of all processed imsets.

For BRIGHTOBJ mode data, dark subtraction is skipped by default in **calnica**, because in general the short exposure times should result in insignificant dark current relative to the object signal. In practice there may be bias components with non-zero amplitude (e.g., akin to “shading”) which are present in BRIGHTOBJ mode data. At present, however, there is no standard procedure for removing these. Given the very limited use of BRIGHTOBJ mode for on-orbit science, we will not discuss its reduction further here.

Chapter 4 includes a more detailed discussion of NICMOS dark and bias components, their properties and behavior, including irregularities which are not well handled by the standard processing pipeline, and which require special care on the part of the user. Chapter 4 also discusses recent updates to the dark reference files available from STScI, including the new dark generator WWW tool. Proper removal of additive instrumental signatures (i.e. dark and bias) can be one of the most important steps in achieving high quality, science grade NICMOS data reductions, and we recommend that the user read the relevant sections of chapter 4 in detail.

NLINCORR (Linearity Correction)

The linearization correction step corrects the integrated counts in the science image for the non-linear response of the detectors. The observed response of the detectors can conveniently be represented by 2 regimes:

- At low and intermediate signal levels the detector response deviates from the incident flux in a way that is correctable using the following expression:

$$F_c = (c_1 + c_2 \times F + c_3 \times F^2) \times F$$

- where c_1, c_2 and c_3 are the correction coefficients, F is the uncorrected flux (in DN) and F_c is the corrected flux. In practice the coefficient c_1 is set to 1, so that the total correction increases from a value of 1 starting at the zero signal level.

- At high signal levels—as saturation sets in—the response becomes highly non-linear and is not correctable to a scientifically useful degree; the saturation level is about 30,500 DN, with a standard deviation of about 2,000 DN.

The NLINCORR step applies the linearity correction to pixels with signal below their defined saturation levels. However, it applies no correction to pixels in the high signal regime, but rather flags them in the DQ image as saturated (DQ value = 64). This step uses the NLINFILE reference file, which consists of a set of images containing the c_1, c_2, and c_3 correction coefficients and their variances at each pixel. The `[NODE,2]` extension of the NLINFILE sets the saturation value for each pixel. Error estimates on the correction applied to non-saturated pixels are propagated into the ERR images of all imsets processed. Data quality flags set in the NLINFILE are also propagated into the processed DQ images. There is one NLINFILE per detector.

Early versions of NICMOS non-linearity correction used a linear correction scheme, rather than the 2nd-order parameterization that is now employed.[1] Starting in calnica v3.3, the NLINCORR step was updated to accommodate the 2nd-order correction, but is backwards-compatible such that old NLINFILEs using the linear correction may still be used if desired. New reference files have been created that include these higher order corrections. Additionally, the nonlinearity reference files include a [NODE,1] extension. This sets the data value below which no nonlinearity correction is applied. It now appears instead that the NICMOS arrays are somewhat nonlinear at all count levels. In the new NLINFILEs, therefore, the [NODE,1] values are uniformly set to 0.0. Because calnica v3.3 was released after the end of NICMOS Cycle 7 operations, all Cycle 7 and 7N data retrieved from the HST archive prior to 26 September 2001 were processed by OPUS using the older non-linearity corrections. If you think that your data may benefit from the newer, more accurate linearity corrections, you should reprocess the images (see section 3.5), or retrieve them again using OTFR, which will automatically process them using the new nonlinearity corrections.

[1] The *correction* term in the nonlinearity equation given above is quadratic. This is then multiplied by the uncorrected flux, yielding an effectively cubic relation between uncorrected and corrected values.

BARSCORR (Bars Correction)

Some NICMOS images will have pairs of bright and dark columns or rows, which have come to be known as “bars”. The bars are believed to arise from electrical cross-talk in the detector lines during the readout of one camera when another of the cameras enters the auto-flush idle state. The bars manifest themselves as a noiseless DC offset of a few DNs along a pair of columns or rows, with the pattern replicated exactly in all four image quadrants. They are discussed and illustrated in section 4.2 of this *Handbook*.

Versions 3.3 and higher of **calnica** use the BARSCORR routine to remove the effects of the bars from MULTIACCUM observations. The routine scans pairs of columns or rows (depending on the camera) in each readout of the MULTIACCUM observation and identifies those with median signals more than 2σ different from the surrounding columns or rows as containing a bar. The user can set a different bars detection threshold by using the `barthresh` task parameter for **calnica**. It flags these pixels with a data quality value of 256 (bad pixel detected during calibration) in the DQ array of the appropriate imsets. In the subsequent CRIDCALC calibration step, where the data from all readouts is combined, the flagged pixels are rejected, so that the final combined image (`*_cal.fits` file) will be free of the bars.

No reference file is used by this step.

BARSCORR is available only in version 3.3 and higher of calnica, which was released after the end of NICMOS operations and instrument warm-up. Therefore, all Cycle 7 and 7N NICMOS data retrieved from the HST archive before 26 September 2001 were processed without the BARSCORR step. In order to take advantage of this step, you will need to recalibrate your data, or to retrieve them again from the Archive using OTFR.

FLATCORR (Flat Field Correction)

In this step the science data are corrected for variations in gain between pixels by multiplying by an (inverse) flatfield reference image. This step is skipped for observations using a grism because the flatfield corrections are wavelength dependent. This step uses the FLATFILE reference file, which contains the flatfield image for a given detector and filter (or polarizer) combination. Error estimates and DQ flags contained in the FLATFILE are propagated into the processed images. There is one FLATFILE per detector and filter combination.

UNITCORR (Convert to Count Rates)

The conversion from raw counts to count rates is performed by dividing the science (SCI) and error (ERR) image data by the exposure time (TIME) image data. No reference file is needed.

PHOTCALC (Photometric Calibration)

This step provides photometric calibration information by populating the photometry keywords PHOTMODE, PHOTFLAM, PHOTFNU, PHOTZPT, PHOTPLAM, and PHOTBW with values appropriate to the camera and filter combination used for the observation. The photometry parameters are read from the PHOTTAB reference file, which is a FITS binary table containing the parameters for all observation modes. The values of PHOTFLAM and PHOTFNU are useful for converting observed count rates to absolute fluxes in units of erg/s/cm^2/Angstrom or Jy, respectively (see section 5.3). PHOTCALC does not alter data values (which remain in units of counts or counts per second), but simply populates header keywords with the appropriate calibration information.

CRIDCALC Cosmic Ray Identification and Signal Accumulation)

This step identifies and flags pixels suspected of containing cosmic ray (CR) hits. For MULTIACCUM mode observations, this step also combines the data from all readouts into a single image. In MULTIACCUM mode, the data from all readouts are analyzed pixel-by-pixel, iteratively computing a linear fit to the accumulating counts-versus-exposure time relation and rejecting outliers from the fit as CR hits. The default rejection threshold is set to 4σ, but the user can override this, if desired, by setting the crthresh task parameter for **calnica**. The fit for each pixel is iterated until no new samples are rejected. Pixel samples identified as containing a CR hit are flagged in the DQ images of the intermediate MULTIACCUM (*_ima.fits) file, with a DQ value of 512. The pixel values in the SCI and ERR images of the _ima file, however, are left unchanged.

Once all outliers have been identified, a final countrate value, and its uncertainty, are computed for each pixel using only non-flagged samples. The result of this operation is stored as a single imset in the output *_cal.fits file in which the number of unflagged samples used to compute the final value for each pixel and the total exposure time of those samples is reflected in the SAMP and TIME images, respectively. The variance ascribed to the final mean countrate is the uncertainty in the slope of the counts-versus-time relation at each pixel location, and is recorded in the ERR image of the *_cal.fits file. Pixels for which there are no unflagged samples, e.g., permanently hot or cold pixels, will have their output SCI, ERR, SAMP, and TIME values set to zero, with a DQ value that contains all flags that were set.

CRIDCALC is only applied to MULTIACCUM images. For data taken in ACCUM or BRIGHTOBJ mode, both the raw and calibrated images will contain cosmic rays and should be treated as with ordinary CCD data.

BACKCALC (Predict Background)

This step computes a predicted background (sky plus thermal) signal level, based on models of the zodiacal scattered light and the telescope plus instrument thermal background. This step uses the BACKTAB reference table which contains the background model parameters. Results of these predictions, along with direct estimates of the background level from the data themselves, are written to the BACKEST1, BACKEST2, and BACKEST3 header keywords. The image data are not modified in any way. **At the time of this writing, this step has not yet been implemented.** If there are future changes to the calibration procedures or software regarding the BACKCALC step, these will be reported in the *Space Telescope Analysis Newsletter (STAN)* and posted on the NICMOS website.

WARNCALC (User Warnings)

In this step various engineering keyword values from the `*_spt.fits` files are examined and warning messages are generated if there are any indications that the science data may be compromised due to unusual instrumental characteristics or behavior. **At the time of this writing, this step has not yet been implemented.** Any future changes to the software implementing WARNCALC will be reported in the *Space Telescope Analysis Newsletter (STAN)* and posted on the NICMOS website.

3.4 Mosaicing: calnicb

Observing strategies with NICMOS vary according to the nature of the target object and of the wavelength chosen for the observation. Extended objects may require mosaicing. Long wavelength observations will need chopping onto the sky to remove the telescope thermal background from the target frame. Multiple repetitions of the same exposure may be requested to improve cosmic ray removal, to control statistical fluctuations, and to increase the signal-to-noise on one target while avoiding saturation on another. Dither (mosaicing) and chop patterns of exposures are specified at the Phase II proposal level via the optional PATTERN parameter; multiple exposures at the same pointing are specified in Phase II by setting the Number_of_iterations to a value greater than one. All these options (which can also be set simultaneously) create an *association* of datasets (see the discussion of "Associations" in appendix B).

The **calnicb** task produces the combined, or *mosaiced*, image from the multiple images contained in a NICMOS association. The task also performs background subtraction and source identification on the images in the association.

It should be noted that **calnicb** is not the only method available for creating mosaics from multiple NICMOS images, nor is it necessarily ideal for all applications. It was designed to implement the potentially complex task of image mosaicing in a pipeline fashion, but sometimes a more careful, manual treatment can be worthwhile. **Calnicb** does not include any correction for geometric distortion in NICMOS images (which is small, but may be important for some applications; see section 5.4 for a discussion), nor does it allow pixel subsampling which may be useful in order to improve image resolution for undersampled images (especially with NIC3). Also, some NICMOS users dithered their observations "manually" using POS-TARG offsets rather than the canned patterns available in Phase II; such data sets will not be linked as associations in the HST archive. Finally, you may wish to have more interactive control over the various stages of the background subtraction, registration, and coaddition process than **calnicb** offers. When analyzing NICMOS data, you may find it useful to explore other means of combining dithered exposures into a final image mosaic, such as the **drizzle** routine and associated software available in the **stsdas.analysis.dither** package (see also section 5.4.4). For many applications, however, **calnicb** will produce excellent results.

The HST Dither Handbook (Koekemoer et al. 2002), provides extensive information about the software in the stsdas.analysis.dither package. The Handbook has many examples showing how to use the drizzling rountines, including one fully worked-out exercise concerning drizzling with NICMOS data.

3.4.1 Input Files

Three pieces of input data are needed by **calnicb**:

1. The *association table* (*assoc_id*`_asn.fits`): this is a table containing the list of members in the association and relevant information on the association type, as given in table 3.1

Table 3.1: Columns of the Association Table (input to calnicb):

Column Name	Meaning
MEMNAME	Rootname (*IPPPSSOOT*) of each image in the association.
MEMTYPE	Role or type of each member: EXP-TARG = input exposure for target EXP-BCK*n* = input exposure of *n*'th background (for chop patterns) PROD-TARG = output product containing target PROD-BCK*n* = output product containing n'th background (for chop patterns)
MEMPRSNT	Flag indicating whether or not a member is present (needed by the STScI automatic pipeline processing).

The table extension header of the *assoc_id*`_asn.fits` file also contains the keywords which control the background illumination pattern correction (ILLMCORR). The keywords used are: ILLMCORR (whether or not the correction is to be performed) and ILLMFILE (reference file name for the illumination correction). These are explained in chapter 2, and discussed further in section 3.4.3.

2. The input calibrated images (*ippssoot*`_cal.fits`): the science data images which are part of the association, as listed in the first column of the association table. The images are usually the calibrated outputs of **calnica**.
3. The support files (*ipppssoot*`_spt.fits`), containing engineering information, so that **calnicb** can transfer this information to the output support files.

3.4.2 Output Files

Calnicb produces three types of output:

1. An updated copy of the association table (*assoc_id*`_asc.fits`): this copy of the *assoc_id*`_asn.fits` file contains additional information about the processing that took place. The *assoc_id*`_asc.fits` file contains four additional columns, listed in table 3.2

Table 3.2: Additional Columns of the output Association Table:

Column Name	Meaning
BCKIMAGE	Flag indicating whether or not the image was used to compute the background.
MEANBCK	Values of the mean background for the image (DN/sec).
XOFFSET	X-offset (in pixels) of the image from the reference frame; a positive value means a positive offset of the image (not of the sources) relative to the reference.
YOFFSET	Y-offset (in pixels) of the image from the reference frame; a positive value means a positive offset of the image (not of the sources) relative to the reference.

Additional information contained in the header of the *assoc_id*`_asc.fits` table is the MEAN_BKG keyword, which gives the constant background signal level subtracted from all images in the association.

2. One or more output mosaic images (*assoc_idn*`_mos.fits`): the number of output mosaic images depends on the pattern. The target field is always contained in the *assoc_id*`0_mos.fits` file. Patterns which involve chopping onto the sky to produce background reference images result in multiple *assoc_idn*`_mos.fits` files after processing through **calnicb**, with the background positions identified by *n*=1 to 8.

3. One *assoc_idn*_spt.fits support file for each *assoc_idn*_mos.fits file created.

3.4.3 Processing

The basic philosophy of the **calnicb** algorithm is to remove the background from each image after source identification, to align the images by calculating offsets, and to produce the final mosaic. The processing steps of **calnicb** can be summarized as follows:

1. Read the input asn table and input images.
2. Determine processing parameters from keyword values.
3. Combine multiple images at individual pattern positions.
4. Identify sources in the images.
5. Estimate and remove the background signal.
6. Create a mosaic image from all pattern positions.
7. Write the output association table and mosaic images.

The sections below discuss steps 2 through 6 in greater detail.

Processing Parameters

Header keywords from the input *_cal.fits images are read and evaluated in order to guide the **calnicb** processing. One set of keywords (table 3.3) pertains to the association as a whole and therefore are read only once from the first input member image:

Table 3.3: Keywords Common to all Datasets in an Association

Keyword	Purpose
INSTRUME	Check whether they are NICMOS data.
CAMERA	Camera number.
FILTER	Filter name; if set to “BLANK”, the association is made of darks.
IMAGETYP	Type of image (EXT=external, DARK=dark frames, FLAT=flat-field images).
NUMITER	Number of iterations for each exposure.
PATTERN	Pattern used.
NUMPOS	Number of independent positions in the pattern.

A second set of header keywords (table 3.4) are specific to each member of the association, and must be read from each input image.

Table 3.4: Dataset-specific Keywords

Keyword	Purpose
PATT_POS	Position of the image in the pattern.
BACKEST*n*	Background estimates from **calnica.**
CRPIX*n*, CRVAL*n,* CD*n_n,* CTYPE*n*	World Coordinate System (WCS) information (see table 2.3).

Based on this information, an inventory is taken of which input images exist, where they belong in the pattern, how many images there are at each pattern position, which images belong to the target field, which ones are from background fields, and to which output mosaic image each input image will contribute. As part of the input process, the appropriate ILLMFILE reference file is loaded. Note, however, as is discussed below under “Background Estimation and Removal,” the ILLMFILEs used in this step are dummies, and have no effect on the data. In fact, the ILLMFILEs provided by STScI have their PEDIGREE set to DUMMY, thus forcing the stage to be skipped. The dummy reference files *must* be present, however, in order for **calnicb** processing to proceed.

Combination of Multiple Exposures

If there is more than one image at any pattern position (NUMITER > 1), the multiple images at each position are first registered and then combined into a single image. The coordinates (as determined by the WCS keywords) of the first image at a given pattern position are used as a reference for the registration. The offsets to all other images at that pattern position are first computed by comparing their WCS data, and then refined using a *cross-correlation* technique**,** down to a level of 0.15 pixels. The cross-correlation technique employs an algorithm which minimizes the differences between fluxes in the images. The computed offsets, in units of pixels, are recorded in the output association table. After determining the relative offsets, the images are aligned using *bilinear interpolation* and are then combined on a pixel-by-pixel basis. The combined pixel values are computed as a weighted mean of all unflagged (i.e., DQ = 0) samples, using the input image ERR values as weights. If three or more samples are present, iterative σ-clipping is performed to reject outliers. The number of samples used at each pixel and the total integration time are retained.

Source Identification

The source identification step is used for excluding sources when the background in the images at each pattern position is estimated. The images at each pattern position are searched for pixels suspected to contain signal from a source. The median signal level in the image is computed and pixels that are more than 4.5σ above the median are considered as candidates. Spurious results, such as pixels containing cosmic-ray hits, are filtered out by searching neighboring pixels and only retaining those candidates that have two or more neighbors that are also above the threshold. The DQ flag of the source-affected pixels is then set to 1024.

Background Estimation and Removal

The background signal is estimated and removed from the images at each pattern position. Two types of background are subtracted from the images:

1. A *constant background* signal level, which is estimated from the images themselves.
2. In principle, a two-dimensional residual background may exist due to spatial variations in the thermal emission of the telescope and instrument. **Calnicb** has a mechanism for removing this by subtracting the ILLMFILE reference image from each image. In practice, however, it does not appear that there are strong spatial illumination variations requiring such corrections, and therefore the ILLMFILEs used in the processing pipeline are dummies, with no effect on the data values. The step may also be turned off entirely by setting ILLMCORR to OMIT in the association table (by default, it is set to PERFORM, even though dummy files are used). In order to disable ILLMCORR, the keyword must be changed in the association table header itself, *not* in the primary image header of the association table FITS file. This must be done with the **tables.parkey** task, not with **hedit**. See the highlighted note at the end of section 3.5.2 for how to do this. Ordinarily, it is simplest to just leave ILLMCORR=PERFORM and use the dummy reference files.

The *constant background* signal level is estimated and removed as follows.

1. With chop patterns, the median and average deviation of the signal in the image at each chop position is computed. In addition to excluding bad and source-flagged pixels, the calculation of the median also uses iterative sigma-clipping to reject outliers.
2. With dither-only patterns, or with multiple-exposure single pointings, the median and average deviation of each target image is computed. The result for each image is compared to the background estimate provided by **calnica**, which is (in principle) stored in the BACKEST1 header keyword of each image. The value computed by **calnicb** is accepted if it is less than 5σ deviant from that of **calnica**, otherwise the **calnicb** value is assumed to be biased by the presence of sources and the **calnica** value is substituted for it. Note, however, that normally the value of BACKEST1 would be populated by the BACK-

CALC step of **calnica**, which has never been implemented (see section 3.3). Therefore ordinarily BACKEST1 has value 0.0, and in this case the background value computed by **calnicb** is always used, unless the user has manually set BACKEST1 to some non-zero value.

3. The global constant background signal is computed by taking the mean of the background values for each image, again using iterative sigma-clipping to reject outliers.
4. The final mean background value is subtracted from all images (both target and background images, if present).
5. With dither-only patterns, the user has the option of subtracting the individual background values computed for each image from themselves, rather than computing and subtracting a global mean background value. This option is controlled by the `meanbkg` task parameter for **calnicb**. The default value, "yes", indicates that the global mean is to be used.

Mosaic Construction

Mosaic (MOS) images are created for each independent pointing within the pattern. For example, a combination DITHER-CHOP pattern will produce one mosaic image out of the dithered pattern at each CHOP location on the sky. Each mosaic image is created as follows:

1. The relative offsets between images within the mosaic are computed from their WCS information and refined using cross-correlation (as in the case of multiple exposures at each pattern position; see "Combination of Multiple Exposures" on page 3-20). The first image in the list for each mosaic is used as a reference image.
2. An empty mosaic image is created with *x* and *y* dimensions large enough to encompass the maximum offsets in each direction.
3. Pixel values in the mosaic image are populated by combining samples from overlapping images. The individual images are aligned using bilinear interpolation and the value at a given mosaic pixel location is computed from the error-weighted mean of the samples at that location. Samples flagged as bad are excluded and, if three or more samples are present, iterative sigma-clipping is used to reject remaining outliers. The number of samples retained for a given pixel and their total integration time is recorded in the SAMP and TIME images, respectively. If all samples are rejected for a pixel, the mosaic image SCI, ERR, SAMP, and TIME values are set to zero and a combination of all DQ flags is retained.

3.5 Recalibration

This section is intended to help you decide whether your data were calibrated with optimal calibration reference files and to help you decide whether you need to recalibrate your data.

3.5.1 Why Recalibrate?

In some cases, NICMOS calibrated data produced by the OPUS pipeline—the standard calibration—are adequate for scientific applications. However, as with all instruments, there is often room for improvement, and you may find it worthwhile or even necessary to reprocess your data, perhaps following additional procedures which are not part of the standard pipeline. As described at the start of this chapter, Cycle 7 and 7N data delivered to the observer or retrieved from the HST archive before 26 September 2001 were calibrated with the best, instrument configuration-specific reference files available at the time of the observation. However, updated or timely reference files sometimes do become available after the data were taken and first processed. As one important example, the DARK refrerence files were substantially updated and improved during and after Cycle 7. Improved software for calibration (e.g., updates to both **calnica** and **calnicb,** as well as new tasks external to the standard pipeline) may also become available as our understanding of the instrument performance increases with experience. For example, the ZSIGCORR step of **calnica** was added part way through the lifetime of the instrument, and BARSCORR was implemented in **calnica** version 3.3 released after the end of NICMOS operations. Many data sets retrived from the archive during Cycle 7 have not had the ZSIGCORR step applied, and none have had the BARSCORR correction. Other improvements to the pipeline tasks were developed along the way, and older data sets (particularly those obtained during the first few months of the NICMOS on-orbit operations) will often benefit from reprocessing with the latest software and reference files.

After 26 September 2001, NICMOS data retrieved from the Archive are automatically reprocessed by OTFR using the latest pipeline software and calibration reference files. This is the easiest way to recalibrate your data. However, there are still situations where you may wish to reprocess your data locally, using non-standard reference files or software tools. For example, the NICMOS bias, dark current, and flat field structure are all functions of instrument temperature. STScI has developed new WWW-based tools for generating temperature-specific DARK and FLAT reference files (see section 4.1.5) which may improve the quality of your data reductions. At present, the only way to take advantage of these special darks and flats is to reprocess the data yourself locally, although the

NICMOS group is currently working on implementing temperature-dependent dark correction automatically in the **calnica** pipeline.

Finally, NICMOS data are subject to a variety of anomalies which may complicate the task of data reduction. These are discussed extensively in chapter 4. In many cases, procedures and software for handling these anomalies were not available when the observations were made or retrieved from the Archive. If you notice unusual features in your data (see, e.g., the checklist at the start of chapter 4), or if your analysis requires a high level of accuracy, you may wish to explore whether a better set of calibration reference files exist than those that were used to process your data, or if additional processing steps may be needed. If better files are available or the calibration software has changed significantly, you may choose to recalibrate your data using the new files or software.

Finding that a calibration reference file has changed since your data were calibrated doesn't always mean that you have to recalibrate. The decision depends very much on which calibration image or table has changed, and whether that kind of change to your data is likely to affect your analysis in a significant way. Before deciding to recalibrate, you might want to retrieve the new recommended reference file or table and compare it to the one used to calibrate your data at STScI in order to determine if the differences are important. You can use the table tools in the IRAF **ttools** package to manipulate and examine calibration tables. Reference files can be manipulated in the same way as your science data.

Finally, the observations may have been made in a non-standard way. Some of the input files (e.g. `*_asn.fits`) may require manual editing before recalibration.

3.5.2 Recalibrating the Data

This section describes the mechanics involved in actually recalibrating a dataset. As noted above, the simplest way to recalibrate your data is to retrieve it again from the HST Archive using OTFR. However, it is sometimes more convenient to simply reprocess your raw data locally using the latest pipeline software or reference files. In some cases, when you may wish to use special, customized reference files or processing software, local reprocessing is your only option.

The basic steps in recalibrating a dataset on your own computer are:

1. Assemble any necessary reference files or tables and your raw data files.
2. Set the desired calibration switches and reference file name keywords in the *primary header* of your raw (`*_raw.fits`) data file. These determine which steps will be executed by the calibration software and which reference files will be used to calibrate the data.

3. Run the calibration software.

Assembling the Input Files

In order to recalibrate your data, you need to retrieve *all* of the reference files and tables that are used by the calibration steps you want to perform. The source of these files is the Calibration Database (CDBS) at STScI. A complete description of how to retrieve the reference files is given in chapter 1 of the HST Introduction.

Setting the Calibration Parameters

The calibration software is completely data-driven, meaning that the calibration steps to be carried out are determined by the values of the *calibration switches* and the *calibration reference files* keywords contained in the primary header of the file to be processed. An important step is then to set the calibration switches and reference file keywords in the primary header of your raw data file (`*_raw.fits`) to reflect how you want the data recalibrated and which reference files you want to use at each step in the process. This is done most easily with the **chcalpar** task in the **hst_calib.ctools** package of STSDAS or with the **hedit** task in the IRAF **images** package.

The calibration switch keywords reside only in the primary header of NICMOS FITS files. Therefore it is critically important to specify extension number zero when passing file names to tasks like hedit to modify these keywords. For example, to modify calibration keywords in the file n3xe01bhm_raw.fits, be sure to use the name n3xe01bhm_raw.fits[0] as input. If you specify any other extension number, the keywords you modify will end up getting written into the header of that extension instead, where calnica will not find them.

The **chcalpar** task takes a single input parameter—the name(s) of the raw data files to be edited. When you start **chcalpar**, the task automatically determines that the image data are from NICMOS and opens a NICMOS-specific parameter set (*pset*) that will load the current values of all the calibration-related keywords. To edit the calibration keyword values:

1. Start the **chcalpar** task, specifying the image(s) in which you want to change keyword values. Note that with **chcalpar**, it is not necessary to append the primary header extension `[0]` to the image name. If you specify more than one image, for example using wildcards, the task will read the initial keyword values from the first image in the list. For example, you could change keywords for all NICMOS raw

science images in the current directory (with initial values from the first image), using the command:

```
ct> chcalpar n*raw.fits
```

2. After starting **chcalpar**, you will be placed in **eparam** [the IRAF parameter editor], and will be able to edit the set of calibration keywords. Change the values of any calibration switches, reference files or tables to the values you wish to use for recalibrating your data.

Exit the editor when you are done making changes by typing `:q` two times. The task will ask if you wish to accept the current settings. If you type “y”, the settings will be saved and you will return to the IRAF `cl` prompt. If you type “n”, you will be placed back in the parameter editor to redefine the settings. If you type “a”, the task will abort and any changes will be discarded.

As delivered from the archive, image header parameters which specify the names of the calibration reference files (e.g, FLATFILE, DARKFILE, etc.) take the form nref$name_ext.fits or ntab$name_ext.fits. The prefixes nref$ and ntab$ are environment variables pointing to the CDBS directories where the reference files reside at STScI. This can be convenient, and you may wish to keep your calibration reference files in a particular directory when reprocessing your data. However, because calnica is a stand-alone C program called by IRAF, the environment variables nref and ntab must be defined at the system host level, before starting an IRAF session. For UNIX environments, this is done with setenv nref /path/ and setenv ntab /path/, where you should specify the path to your reference file directory. Do not forget the trailing slash!

Running the Calibration Software

After you change the header keyword values for your raw data files, you are ready to recalibrate your data. To run **calnica**, type the name of the task followed by the names of the input raw data file and desired output calibrated data file. For example, to recalibrate the dataset `n0g70106t`, you could type:

```
ni> calnica n0g70106t_raw.fits n0g70106t_cal.fits
```

or simply:

```
ni> calnica n0g70106t ""
```

To run **calnicb** the name of the association table must be given as input:

```
ni> calnicb assoc_id_asn
```

To run **calnicb** on a subset of the `*_cal.fits` files, it is sufficient to edit the `*_asn.fits` table and remove the undesired files.

The calibration routines calnica and calnicb will not overwrite an existing output file. If the calibration tasks are run in the directory where the original calibrated files are located, a different output file name must be specified.

There are some calnicb processing parameters which reside in the association (*_asn.fits[1]) table header and not in any FITS file image header, and which therefore cannot be changed using chcalpar or hedit. In particular, to change the ILLMCORR and ILLMFILE parameters, you must use the parkey task on the *_asn.fits[1] file. As an example, to change the ILLMFILE, type: parkey /path/name_ilm.fits data_asn.fits[1] ILLMFILE, where /path/name_ilm.fits is the full name and path to the ILLMFILE, and data_asn.fits is the association table you are editing. In order to set ILLMCORR to OMIT and skip this processing step entirely, type: parkey OMIT data_asn.fits[1] ILLMCORR.

CHAPTER 4:

Anomalies and Error Sources

In this chapter. . .

The previous chapter described the basic stages of NICMOS pipeline processing. As with any instrument, however, high quality data reduction does not end with the standard pipeline processing. NICMOS data are subject to a variety of anomalies, artifacts, and instabilities which complicate the task of data reduction and analysis. Most of these can be handled with careful post-facto recalibration and processing, which usually yields excellent, scientific grade data reductions. Careful NICMOS data processing usually requires a certain amount of "hands-on" interaction from the user, who must inspect for data anomalies and treat them accordingly during the reduction procedures.

This chapter describes the most common problems affecting NICMOS data at the level of frame-by-frame processing. In some cases, recognizing and treating problems with NICMOS data requires a moderately in-depth understanding of the details of instrumental behavior; problems with dark and bias subtraction are good examples. Where appropriate, this chapter offers a fairly detailed discussion of the relevant workings of the instrument, but the reader should consult the *NICMOS Instrument Handbook* for further details.

Each section of this chapter deals with a different aspect of NICMOS data processing, roughly following the order of the processing steps in the standard STSDAS pipeline. Various potential problems are described and illustrated. Each discussion then has a subsection labeled "Cures" which offers possible solutions to the problems at hand. The art of NICMOS data processing is still evolving, and new techniques are being developed by users worldwide. Some of these have been encoded into software which is now distributed with STSDAS. New processing methods and routines are still being developed at STScI and will be provided in the future: the reader should consult the STScI NICMOS WWW pages to look for new developments and software tools.

We begin with a checklist of potential NICMOS instrumental anomalies and potential data processing problems about which the user should be aware. Each of these is discussed in further detail in the sections which follow: the nature of the problem and its impact on NICMOS data is illustrated, and possible processing solutions are considered. The relevant sections for each anomaly are given in parentheses below.

It is expected that some aspects of NICMOS performance and anomalies may be somewhat different in Cycle 11 and beyond with the NICMOS Cooling System. The discussions of data anomalies and appropriate solutions in this edition of the NICMOS Data Handbook are based on the properties and behavior of the instrument during Cycles 7 and 7N. NICMOS users should carefully monitor developments and updates posted on the STScI NICMOS web pages when analyzing data taken in Cycle 11 and beyond.

NICMOS Problems to Watch Out For: A Checklist

- Bias and dark subtraction problems, including:
  - residual shading (section 4.1.2, section 4.1.5)
  - variable quadrant bias or "pedestal" (section 4.1.2, section 4.1.5)
  - bias jumps or bands (section 4.1.2, section 4.1.5)
- Bars (section 4.2)
- Nonlinearity correction uncertainties
  - new nonlinearity corrections (section 4.3.1)
  - non-zero zeroth read correction (section 4.3.2)
  - uncorrected saturation (section 4.3.3)

- Flatfield issues, including color dependent flat fields (section 4.4)
- Pixel defects and bad imaging regions, including:
    - Bad pixels (section 4.5.1)
    - Grot (section 4.5.1)
    - Erratic middle column/row (section 4.5.2)
    - Coronagraphic hole masking (section 4.5.3)
    - Vignetting (section 4.5.4)
- Effects from bright targets:
    - Photon-induced persistence (section 4.6.1)
    - Post-SAA cosmic ray persistence (section 4.6.2)
    - The "Mr. Staypuft" Effect (amplifier ringing) (section 4.6.3)
    - Optical ghosts (section 4.6.4)
- Cosmic rays of unusual size (section 4.7)
- Scattered earthlight (section 4.8)

4.1 NICMOS Dark Current and Bias

Some of the major challenges for achieving high quality NICMOS data reduction arise from difficulties in removing additive components of the instrumental signature that are present in a raw NICMOS image. For the purpose of discussion here, we will divide these additive components into two categories, *bias* and *dark*, according to whether or not the signal is noiseless and purely electronic in origin (bias), or noisy and arising from thermal or luminous sources (dark). In practice, the NICMOS bias and dark signals each consist of several different components which exhibit a range of different behaviors.

In the standard reference files used for processing NICMOS data, dark and bias components are combined together as a single DARK image and are handled in the same step (DARKCORR) of the **calnica** pipeline processing. NICMOS dark images (really dark + bias) are highly dependent on the readout history of the array since it was last reset, and therefore cannot be simply rescaled to the exposure time of the science data (as is done with most conventional CCD data). Each science file must be calibrated with a dark frame of equal exposure time and number of readouts.

Because there is such a large variety of NICMOS MULTIACCUM readout sequences, it was considered to be impractical to obtain and regularly monitor darks in every possible sequence. However, most components of the NICMOS bias and dark current are highly reproducible and can be reliably calibrated. On-orbit darks obtained during SMOV and throughout the lifetime of the instrument have been used to characterize the dependence of the major dark and bias components on pixel position, on time, and on temperature for each of the three NICMOS detectors. This information has been used to construct "synthetic" dark calibration reference files for all MULTIACCUM readout sequences, using as basic data the on-orbit darks obtained during the first part of the Cycle 7 calibration program. These are described in section 4.1.3 below.

Unfortunately, some components of the NICMOS bias have turned out to be unstable or unpredictable, making it difficult or impossible to remove them using the standard reference files. In order to do a good job removing additive dark and bias signatures, it is important to understand their origin and behavior. Here we describe the various components of NICMOS biases and darks in some detail, highlighting their stability or lack thereof, and describing (briefly) how they are incorporated into the standard STScI synthetic dark reference images. In section 4.1.5 below we describe methods and tools for measuring and removing residual dark and bias artifacts from NICMOS images.

4.1.1 Dark Current

Linear Dark Current

The true, thermal dark current is the detector current when no external signal is present. This component grows linearly with integration time:

$$D(x,y,t) = dc(x,y) \times t$$

where *D(x,y,t)* is the observed signal in a given readout, *t* is time since reset, and *dc(x,y)* is the dark current. At the operating temperatures used for NICMOS in Cycle 7, the mean dark current for all three cameras was of order 0.1 e^-/sec. It is likely to be higher in Cycle 11 when operating at warmer temperatures with the NCS. The dark current has some two dimensional structure, and is roughly a factor of two higher in the corners than at the center.

Amplifier Glow

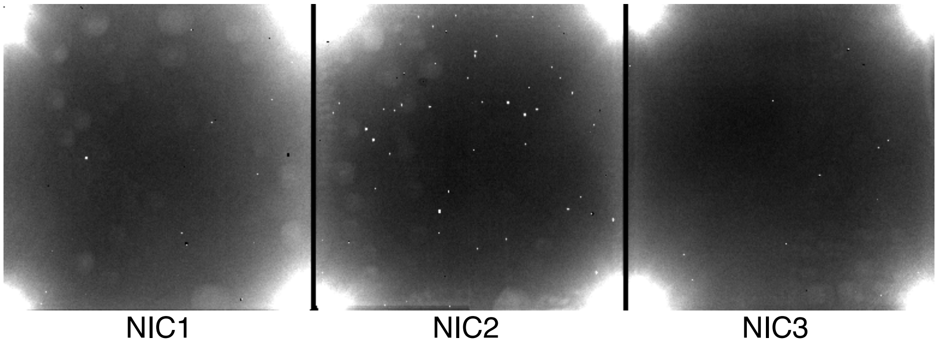
Each quadrant of a NICMOS detector has its own readout amplifier, which is situated close to an exterior corner of the detector. When a readout is made, the amplifier emits radiation which is recorded by the detector, an effect known as amplifier glow (figure 4.1). This signal is largest close to the corners of the detector where the amplifiers are situated, and falls off rapidly towards the center of the array. The signal is only present during a readout, but is repeated for each readout (e.g., a MULTIACCUM sequence or an ACCUM with multiple initial and final reads). Typically the extra signal is about 20-30 DN at the corners of the detector and 2-3 DN at the center, for each readout. The signal is highly repeatable, and exactly linearly dependent on the number of reads. The amplifier glow also depends on the length of time for which the amplifiers are switched on, which is slightly shorter for ACCUM mode.

The amplifier glow is a real signal and is subject to photon statistics, so it is a source of noise in NICMOS exposures. In the processing pipeline and calibration reference files, it is considered to be a component of the dark signal, although its physical origin and temporal dependence is quite different than that of the thermal dark current. Thanks to the repeatability of the signal, images calibrated with the appropriate dark frames (same MULTIACCUM sequence or same exposure time for ACCUM images) will have the amplifier glow removed. For the STScI synthetic dark reference files, the amplifier glow is modeled as

$$A(x,y) = amp(x,y) \times NR$$

where *A(x,y)* is the cumulative signal due to the glow in a sequence, *amp(x,y)* is the amplifier glow signal per readout (a function of the pixel location *(x,y)* and the amp-on time), and *NR* is the total number of readouts of the array since the last reset. In the corners of a full, 26-readout MULTIACCUM response there will be of order 500-800 DN due to amplifier glow, as well as the associated Poisson noise from this signal. This nominal Poisson noise is propagated into the ERR array of the NICMOS calibrated images by **calnica**.

Figure 4.1: Amplifier glow images for NICMOS cameras 1, 2 and 3.



4.1.2 Bias, Shading, and "Pedestal"

There are three readily identifiable (but not necessarily physically distinct) components of the NICMOS bias: the detector reset level, shading, and variable quadrant bias or "pedestal."

Bias Reset Level

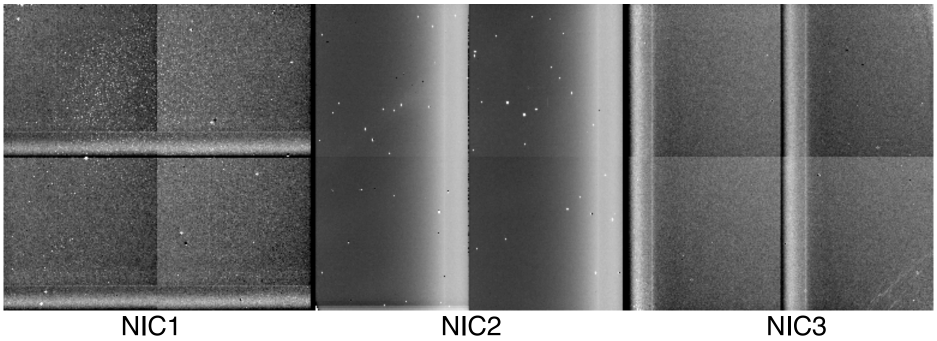
First, a net DC bias with a large, negative value (of order -25000 ADU) is introduced when the detector is reset. This bias is different in each readout quadrant, but is essentially constant within each quadrant. In standard MULTIACCUM processing, this is removed by subtracting the so-called "*zeroth readout*" of the image from all subsequent readouts, e.g. in the ZOFFCORR step of **calnica**. It is therefore not a component of any calibration reference file, but is removed using the zeroth readout of the science image itself.

Shading

Shading is a noiseless signal gradient, a pixel-dependent bias, which changes in the direction of the pixel clocking during a readout. This bias change is caused by a temperature dependence of the readout amplifier bias. The amplifier temperature changes rapidly during the readout of the array. The result is a bias which changes considerably between the time the first and last pixels of a quadrant are read. Visually, this appears as a ripple and a signal gradient across a given quadrant of an uncorrected image (figure 4.2). The amplitude of the shading can be as large as several hundred electrons across a quadrant in NIC2, with smaller amplitudes in NIC1 and NIC3. The shading exhibits all the characteristics of a bias change, including lack of noise.

The shading signal is not the same for each readout, but depends primarily on the time interval since the last readout (not reset) of a pixel. For each readout in a NICMOS MULTIACCUM sequence, this time interval is recorded in the FITS header of each imset by the keyword DELTATIM. If the time Δt between reads remains constant, the bias level introduced by the shading remains constant, but if Δt varies (e.g. logarithmically, as in some MULTIACCUM sample sequences), then the bias level changes with each successive read, and thus the overall shading pattern evolves throughout the sequence.

Figure 4.2: Examples of shading for NICMOS cameras 1, 2 and 3



In addition to the DELTATIM dependence, the shading amplitude and shape also depend on the mean temperature of the detectors, which slowly warmed as the cryogen sublimated over the lifetime of the instrument. Subtle temperature changes during a MULTIACCUM exposure can also lead to shading changes. A sequence with many long DELTATIMEs (such as a SPARS256) can cool between the first and last reads, resulting in a

DELTATIME=256s shading that is different in the 25th read than it was in the 4th read. A similar situation occurs in the MIF sequences, when the DELTATIMEs switch over from the relatively long steps in the middle to the rapid reads at the end: the first short read of the set taken at the end of the observation has slightly different shading than the remaining short reads or the initial short reads.

Numerically, shading is of the form:

$$S(x,y) = s(dt,x,y,T)$$

where the shading *s* is a function of the pixel location, DELTATIME *dt* and detector temperature *T*.

Images of the shading as a function of DELTATIME and temperature can be made by subtracting the amp-glow and linear dark components from DARK observations. These images can then be used to build dark reference files for any sequence given a set of DELTATIMEs and a temperature.

The shading patterns of the NICMOS cameras were measured using data taken early in the Cycle 7 calibration program, and the synthetic dark reference files used in the standard STScI pipeline processing use these shading images (taken with the detectors at ~61.4 K in the fall of 1997). However, as the instrument was monitored over the course of many months, the temperature dependence was discovered. Because the temperature varied during the lifetime of the instrument, so did the shading. This variation is not yet taken into account in the standard pipeline processing done for the HST Archive, and the DARK reference files that have been available from STScI do not currently include temperature dependent shading. Therefore, if a particular observation was taken when the instrument was at a different temperature than that for which the standard DARK reference files were calibrated, then residual errors in shading subtraction may result. The effect is most noticeable for NICMOS Camera 2 because the amplitude of the shading is much larger for that camera. Shading variations are much smaller, and usually negligible, for Cameras 1 and 3. The STScI NICMOS group has developed a fully temperature dependent model for shading in all three NICMOS cameras, and has implemented a WWW-based tool for generating temperature dependent dark reference frames (see "cures," section 4.1.5). As of the time of this writing, the STScI NICMOS group is also working to implement automatic temperature-dependent dark correction in the **calnica** pipeline. It is quite likely that this will be available by early 2002, and readers should check the STScI NICMOS WWW pages and NICMOS STANs for further updates.

Variable Quadrant Bias or "Pedestal"

In addition to the net quadrant bias introduced at array reset, there is some additional offset which is time-variable and, to some degree, stochastic. This variable quadrant bias has been described as the "*pedestal* *effect*" in many discussions of NICMOS data, although we note here that the term "pedestal" has also been applied to other aspects of NICMOS array behavior. The variable quadrant bias is usually constant over a given array quadrant, but different from one quadrant to another. Its amplitude varies from readout to readout, sometimes drifting gradually, but occasionally with sharp changes from one readout to another (not always seen in all quadrants simultaneously).

On 22 August 1997, a modification was made to the NICMOS flight software which reduced but did not eliminate the pedestal effect. Data taken before that date is, in general, severely affected by variable bias levels, and requires careful handling in order to achieve high quality data reductions. However, essentially all NICMOS data, even after the flight software change, are impacted by pedestal to one degree or another.

The variable quadrant bias has two major effects on NICMOS MULTIACCUM data. The first (and generally less important) effect is that the signal in a given pixel, which should normally accumulate linearly with time over the course of an integration (after other sources of bias and dark current are removed, and when intrinsic array non-linearity is corrected), can instead vary irregularly as the bias level in a quadrant changes "underneath" the astronomical signal from source + background. The CRIDCALC step of the **calnica** pipeline fits a linear ramp (counts vs. time) to the accumulating signal in the MULTIACCUM to derive the source + background count rate, with a rejection procedure designed to eliminate transient cosmic ray events (see section 3.3 and section 4.7). A varying bias level can improperly trigger the CRIDCALC cosmic ray rejection or reduce its sensitivity to real cosmic ray events.

Secondly, the net bias change over the course of the exposure results in an additive offset (different in each quadrant) when the MULTIACCUM sequence is reduced to a single count rate image (the `*_cal.fits` file) by CRIDCALC. When the image is flatfielded, this undesired, additive offset is then modulated by the flatfield, and appears as an inverse flatfield pattern in the final, reduced data. For illustration, consider an image where the incident astronomical flux (sources plus sky background) is given by $S(x,y)$. This is modulated by the spatially dependent quantum efficiency, or flatfield, $Q(x,y)$. To this is added a quadrant bias offset B_q, which may be different in each quadrant. Here we neglect all other sources of bias and dark current, assuming that they can be adequately removed by standard processing. The recorded raw image is $I(x,y)$:

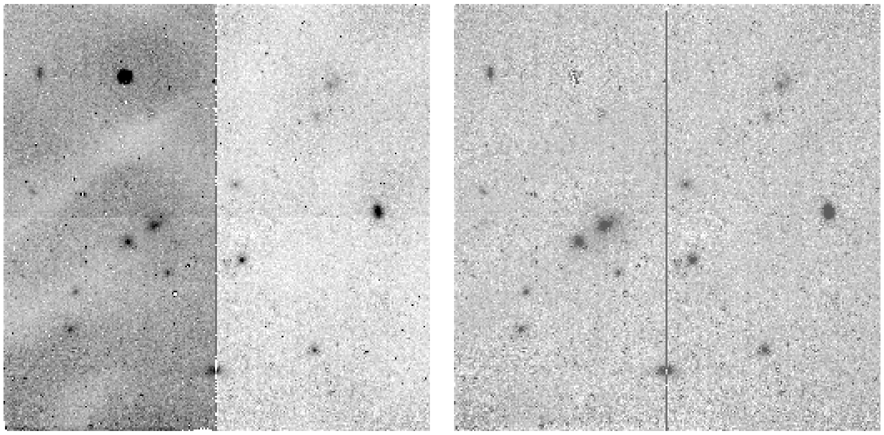
$$I(x,y) = S(x,y) \times Q(x,y) + B_q$$

If this image were then divided by the flatfield (or, to follow the STScI pipeline convention, multiplied by the inverse flatfield), the result would be:

$$I(x,y) * Q^{-1}(x,y) = S(x,y) + B_q \times Q^{-1}(x,y).$$

Thus, the desired image *S(x,y)* is recovered, but an additive, inverse flatfield pattern is also present, with an amplitude that may be different for each quadrant. These inverse flat patterns, along with discontinuities between quadrants, are the typical hallmarks of a pedestal problem in processed NICMOS data (see example in figure 4.3).

Figure 4.3: Data affected by variable quadrant bias. **Left**: image processed normally with **calnica**; note the quadrant intensity offsets, and also the residual flat field pattern imprinted on the data, due to the unremoved bias being multiplied by the inverse flat. **Right:** image after processing through **pedsky**.



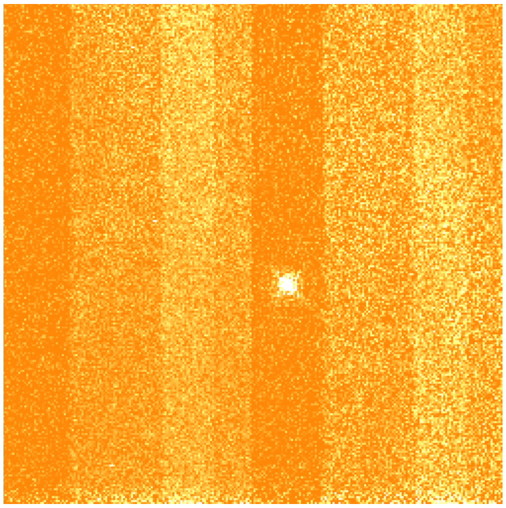
It is important to note here that a residual flatfielding pattern may also arise from reasons completely unrelated to pedestal. In particular, the NICMOS flat fields have a strong color dependence, and the spectrum of the background (especially at longer wavelengths where thermal emission dominates) does not necessarily match that of the lamps used to create the flat fields. Residual patterns may therefore sometimes result from division by standard internal lamp flats, again especially at longer wavelengths in the medium and broad band filters. We return to this point in section 4.1.5 in the discussion of the **pedsky** software routine and again in section 4.4.2. Unremoved shading also introduces a bias offset (but a positionally dependent one) which, when multiplied through by the inverse flatfield, will create a pedestal-like effect.

The unpredictable nature of this variable quadrant bias means that it is not possible to remove it with standard reference frames. (In passing, we note that it also considerably complicates the task of generating "clean" calibration reference files of any sort in the first place.) The user must attempt to determine the bias level from the data themselves and subtract it before flatfielding the data. The difficulty, then, is determining the bias level independent of the sky + source signal present in the data. No one method has been developed which does this equally well for all types of NICMOS data. The methods which have been tried depend on the nature of the target being observed, e.g. sparse fields consisting mostly of blank sky are treated differently from images containing large, extended objects or crowded fields. We discuss pedestal removal techniques in section 4.1.5 below.

Bias Jumps or Bands

Occasionally, spatial bias jumps (sometimes called *bands*) are seen in NICMOS images (figure 4.4). These are apparently caused by a bias change when the amplifiers of one NICMOS camera are being used at the same time as another is reading out. They are very commonly seen in the last readout of a MULTIACCUM sequence, but may occasionally occur in intermediate readouts as well.

Figure 4.4: Bias jumps or "bands" in a NICMOS image.



4.1.3 Dark Reference Files

Synthetic Darks

The NICMOS dark calibration reference files which are used in the HST Archive pipeline processing are so-called "synthetic" dark images. These include the linear dark current, amplifier glow, and bias shading terms described above. The synthetic darks were constructed using on-orbit measurements of the linear dark current and amplifier glow, plus an empirical shading model based on a fit to on-orbit data.

Random Uncertainties (i.e., noise) in the Synthetic Darks

In the center of the NICMOS arrays, where the effects of shading and amplifier glow are smallest, the uncertainties in the dark reference files are dominated by the readout noise. The older STScI synthetic darks were typically based on an average of about 15 measurements per readout sample per pixel. Therefore the estimated pixel-to-pixel uncertainties in the DARK reference files are of the order of 1 DN (about 5 electrons). In the corners of the arrays the amplifier glow is the largest source of noise, increasing as a function of the number of readouts. For the largest number of readouts (26) the estimated uncertainty is of the order of 5 DN (about 27 electrons). It is important to note that the effect of these "random uncertainties" in the calibration files on science data is *not* actually "random," however. The pixel-to-pixel noise pattern in the DARK reference files is systematically imprinted on all science images from which they are subtracted. This can introduce a sort of "pattern noise" in the images, which is apparently random but actually affects the pixel-to-pixel statistics of reduced data in a systematic way. In general, this is not a limiting source of noise in NICMOS data, but it can set a limit to the pixel-to-pixel noise achievable with images reduced by **calnica** using the standard reference files. In the newer, temperature-dependent synthetic dark reference files (now available via a WWW-based tool - see section 4.1.5 below), a much larger number of dark exposures has been averaged to produce the final product, thus reducing this pixel-to-pixel component of the dark frame "noise" to a negligible level.

Systematic Uncertainties in the Synthetic Darks

The dark current pedestal adds some uncertainty to the synthetic darks, since on-orbit dark frames are used to generate the calibration reference files. In essence, the pedestal makes it difficult to establish the absolute DC level of the dark current. However, every effort was made to minimize the effects of the pedestal when making the reference files currently in the database.

Also, during the lifetime of the instrument, no temperature dependence was included in the dark files used in the calibration data base. As discussed elsewhere (see section 4.1.2 and section 4.1.5), this can lead to systematic bias subtraction errors, primarily due to problems with the shading correction.

New dark reference files for all cameras have been generated at STScI which implement a temperature-dependent shading correction. These new files also average a larger number of individual frames for the linear dark current measurements, providing better signal-to-noise for the DARK reference files. The new darks are not yet used in standard Archive OTFR, but can be obtained using the new WWW-based temperature-dependent synthetic dark generation tool (see section 4.1.5 below). Users must then reprocess the data locally using these darks. Initial tests show that using these new darks can significantly reduce pixel-to-pixel noise in processed data and improve the shading correction.

On Orbit Darks

All dark images taken with NICMOS are available through the HST Archive. Certain MULTIACCUM sequences have extensive collections of on-orbit dark data, particularly those used for the dark monitoring program. Other NICMOS MULTIACCUM sequences, however, have very little on-orbit data. Users wishing to use on-orbit dark data instead of synthetic darks may retrieve darks for the camera of interest with the appropriate MULTIACCUM sequence and the correct number of readouts (NSAMP), process them by subtracting the zeroth readout (ZOFFCORR), and combine them on a readout-by-readout basis with some suitable rejection scheme to eliminate cosmic rays (e.g. median or sigma clipping).

In practice there are several difficulties when doing this. First, on-orbit darks are affected by pedestal effects. Care must be taken when averaging frames, particularly with sigma rejection schemes, since the DC bias level of a given quadrant in a given readout may vary considerably from image to image. The average dark image will still have some mean pedestal value in it. Second, one should be careful to examine all dark frames used for the average, and to discard images which are adversely affected by bright object and SAA-induced persistent signal (see section 4.6.2 below). Finally, because shading is temperature dependent, care should be taken to combine darks taken at the same detector temperature (within approximately 0.1 degrees K) as the observations for which they will be used.

Detector temperatures are stored in the NDWTMP11 and NDWTMP13 keywords in the *_spt.fits files. For Cameras 1 and 2, use NDWTMP11; for Camera 3, use NDWTMP13. (These keywords are incorrectly labelled in some NICMOS *_spt.fits headers.)

ACCUM and BRIGHTOBJ Mode Darks

In principle, ACCUM mode allows the user to specify any of a large number (173, to be precise) of possible exposure times, ranging from 0.57 to 3600 seconds, and either 1 or 9 initial and final readouts (NREAD). As was discussed above, the various components of the DARK reference files (e.g., bias shading, linear dark current, and amplifier glow) depend not only on the integration time, but on the number of readouts and the readout delta time intervals. Therefore each and every combination of ACCUM exposure time and NREAD requires a unique dark image for calibration, and it was not practical to calibrate all of these on orbit. In addition, as has also been noted, the shading (particularly for NIC2) also depends on the instrument temperature.

At the present time, there are no "standard" DARK calibration reference files available from the HST Calibration Database for use with ACCUM mode data. The DARK reference files used for processing ACCUM mode images in the OPUS pipeline were dummies. In principle it should be possible to create synthetic ACCUM dark reference files using a procedure similar to that which has been used for MULTIACCUM data. In the future, STScI may make a tool available for generating synthetic ACCUM mode darks, but the analysis needed to do this has not yet been carried out. Many individual, on-orbit ACCUM mode dark exposures are available from the Archive, and it is not unlikely that for any given ACCUM mode science exposure there will be darks available with the right exposure time (if not necessarily the right temperature). If you need to calibrate ACCUM mode science images, you should search the Archive to see if suitable darks are available, or discuss the matter with your Contact Scientist.

In general, BRIGHTOBJ mode exposures are so short that true linear dark current is negligible. Moreover, by design they are generally used only for very bright targets, and most sources of dark and bias are probably relatively unimportant compared to the object signal. The behavior of the detector bias in BRIGHTOBJ mode has not been systematically characterized at STScI, and no DARK reference frames are available for this mode.

4.1.4 What is Removed by Standard Pipeline Processing

The DC bias or reset level is removed from NICMOS images by subtracting the zeroth readout from subsequent readouts. For MULTIACCUM data, this is accomplished by **calnica** with the ZOFFCORR step. For ACCUM data, the zero read subtraction is performed automatically on board the telescope.

The DARKCORR step of **calnica** subtracts a dark reference image, specified by the DARKFILE header keyword, from the science exposure. For MULTIACCUM data, this is done for each sample readout (i.e., imset), matching the appropriate imsets of the DARK reference file to those of the science image. The DARKFILE includes all three additive components described above, i.e., true dark current, amplifier glow, and bias shading.

In general, the DARKCORR step of **calnica**, using standard STScI reference files, should do an adequate job of removing linear dark current, amplifier glow, and most shading.

4.1.5 Cures: How To Get Rid of What's Left

Several components of the bias may not be adequately removed by the standard pipeline dark correction. In particular, residual shading in NIC2 data, bias drifts and jumps, and the net pedestal may still be present after standard processing. Here we describe ways of handling each of these.

Residual Shading

As described above, the bias shading function is temperature dependent, and may not be completely removed from data using the standard synthetic dark reference files. This is particularly true for Camera 2 data, where the shading amplitude is largest. The STScI NICMOS group has now numerically modeled the temperature dependence of the shading and has made a temperature-dependent dark generating tool available via the WWW. This program that can generate synthetic darks for any NICMOS MULTIACCUM sequence and at any temperature within the NICMOS operating range.

When reprocessing NIC2 images, the use of temperature-dependent darks generated with the STScI WWW tool will often significantly improve the quality of the reduced data, minimizing both shading residuals and pixel-to-pixel noise for non-background limited data (i.e., almost all NIC2 images taken through filters at wavelengths shorter than 1.8 microns). You may extract the appropriate detector temperature information from the NDWTMP11 and NDWTMP13 keywords in the *_spt.fits files. Use NDWTMP11 for NIC1 or NIC2, and NDWTMP13 for NIC3.

Variable Quadrant Bias or "Pedestal"

Variable quadrant bias or "pedestal" is not removed by standard processing. STScI has distributed several STSDAS tools for removing this variable bias level. Here we briefly describe the tools that are currently available in the **stsdas.hst_calib.nicmos** package.

biaseq

The **biaseq** task in the **stsdas.hst_calib.nicmos** package is designed to remove the changes in quadrant bias level from readout to readout during the course of a MULTIACCUM exposure. It adjusts the bias levels in each NICMOS quadrant so that the net counts in that quadrant increase linearly with time. This "bias equalization" procedure does ***not*** remove the net bias offset that produces the pedestal effect. Essentially, it removes any temporally non-linear components of the bias drift, i.e., the second and higher order time derivatives of the bias, but leaves an unknown linear term in the bias drift (i.e., the first time derivative). The program cannot distinguish between this linear bias drift and an actual, linearly accumulating astronomical signal, and thus leaves the linear bias term in the data so that it must be removed by some other method (see “pedsky” and “pedsub” below). In principle, **biaseq** will work on any NICMOS MULTIACCUM image, regardless of the nature of the astronomical sources present, provided that there are enough MULTIACCUM samples available. As a by-product, **biaseq** can also attempt to identify and remove bias jumps or bands (see section 4.1.2 above) from individual readouts.

The **biaseq** task must be run on an intermediate image file (`*_ima.fits`) which is produced by partially processing a raw NICMOS data set through only the first few processing steps of **calnica**. The pipeline processing steps BIASCORR, ZOFFCORR, ZSIGCORR, MASKCORR, NOISCALC, NLINCORR, DARKCORR and BARSCORR should be performed before running **biaseq**, but not FLATCORR, UNITCORR or CRIDCALC, i.e., the image should not be flatfielded and should be in units of counts (not counts per second). The **nicpipe** task in the **stsdas.hst_calib.nicmos** package provides a convenient way to carry out the partial **calnica** processing needed as preparation for **biaseq** (see example below, and also section 5.1).

The **biaseq** task assumes that the astronomical signal (sky plus sources) accumulates linearly with time, and that any non-linear behavior is due to changing bias levels that are constant within each array quadrant, except perhaps for bias jumps. If these assumptions are not correct, then the task may not work properly. E.g. if the sky background is changing with time, either because it is dominated by variable thermal emission or because of scattered earthlight (see section 4.8), then the routine may not function correctly. The **nicmos** task **pstats** may be used to compute and graph data statistics versus time or readout number, which can help to identify time-varying background levels. Objects which saturate the NICMOS array will also no longer accumulate signal linearly with time. However, in this case, unless a large fraction of the pixels in a quadrant are saturated, there should be no noticeable effect on **biaseq**. Also, residual shading may result in a bias which changes from readout to readout but is not constant across the quadrant, and this may also cause problems for **biaseq**. In particular, residual shading can improperly trigger the bias jump finding algorithm. If there is residual shading in the images, it should be removed with a temperature-dependent shading correction (see “Residual Shading” on page 4-14) before running **biaseq**. If **biaseq** is run without doing this, the user should at least disable the bias jump finding option. The **biaseq** help pages give further information about this task and its parameters.

pedsky

For NICMOS images of relatively blank fields, free of very bright or large sources which fill a substantial portion of the field of view, the **pedsky** task may be used to measure and remove an estimate of the sky background and quadrant-dependent residual bias (or "pedestal"). The task depends on having a large fraction of the image filled by "blank" sky, and thus may not work well for images of large, extended objects or very crowded fields.

Pedsky runs on a single science image (i.e. not on all the separate readouts of a NICMOS MULTIACCUM file). It operates only on the `[SCI,1]` extension, which is appropriate when the task is run on, e.g., the `*_cal`.fits images that are the final product of the **calnica** calibration pipeline. The task's internal algorithms operate on an unflatfielded image, but a fully-calibrated (including flatfielding) image may be used as input, as the task will check the status of the flatfielding (via the value of the FLATDONE keyword in the input image header) and will temporarily remove and, at the end of processing, reapply the flatfield if necessary.[1] Note, however, that the FLATFILE used for the processing *must* be available locally in order to run pedsky. Therefore if you wish to use this task on reduced data taken from the HST Archive, be sure to retrieve the appropriate flatfield reference file as well.

Following the discussion in section 4.1.2 above, let us say that a NICMOS image $I(x,y)$ may be described as

$$I(x,y) = S(x,y) \times Q(x,y) + B_q$$

where $S(x,y)$ is the incident astronomical flux (sources plus sky background), $Q(x,y)$ is the flatfield, and B_q is the quadrant-dependent bias offset. The **pedsky** task works by minimizing the quantity

$$X^2 = \Sigma_{xy} (I_{xy} - S \times Q_{xy} - B_q)^2$$

which is a measure of the total image variance, as a function of the sky level S and four quadrant bias levels B_q. Here we have made the simplifying assumption that the "true" incident flux $S(x,y)$ in a relatively blank-field image can be approximated as a constant sky background S, i.e. that there are no sources present. In real data where there are real sources, the quantity X^2 includes a contribution due to the presence of actual objects in the image, above the assumed uniform, constant sky level S. The impact of these sources is minimized, however, by using sigma-clipped statistics which exclude pixels with strongly deviant values, e.g. those due to actual astronomical sources, bad pixels, etc. Additionally, the user may apply a ring median filter to the image when computing X^2. This can effectively remove compact or point-like sources, and may help the task perform better for moderately crowded fields, but considerably slows the computation speed. The user may wish to experiment by trying **pedsky** both with and without the ring median option. In practice, a certain amount of "source noise" contribution to X^2 is tolerable to the **pedsky** algorithm. It acts as an offset to the amplitude of X^2, but generally has no effect on the location of the minimum value for X^2 relative to S or B_q.

[1] This is a change from earlier versions of the **pedsky** task and earlier editions of this *Handbook*. Previously, **pedsky** required a partially processed image, and would not work on flatfielded data.

Pedsky works in both interactive and non interactive modes. Alternatively, the user can supply a sky value to be subtracted, in which case the remaining quadrant-dependent pedestal is estimated and subtracted. After **pedsky** processing, the remaining standard calibration steps, including flatfielding, can then be easily applied using another call to the script **nicpipe**.

Note that, in principle, any image may be used to represent the spatial structure of the sky, i.e., you do not need to use a standard NICMOS flatfield reference file. In particular, for some NICMOS images the two-dimensional structure of the sky may not exactly resemble that of the flatfield. This may happen for at least two reasons. First, as will be discussed below (section 4.4.2), the NICMOS flatfields are strongly color dependent, and the spectrum of the internal flatfield lamps does not necessarily match that of the sky background, especially at longer wavelengths (> 1.8 μm) where thermal emission begins to dominate over the zodiacal sky. The result is that the spatial structure of the sky may not be quite the same as that of the flatfield, and that the sky multiplied by the inverse flat reference file may have some residual structure which correlates with the flatfield pattern and contributes to the total image variance X^2 measured by **pedsky**. This can confuse **pedsky**, resulting in improper sky and pedestal measurement. This problem is most important for images dominated by thermal background, but may also affect shorter wavelength data, especially for Camera 3 data where the ratio of background counts to quadrant bias offset amplitude is larger than for the other two cameras. Second, at longer wavelengths, the thermal background generated within the telescope may illuminate the detector differently than does the zodiacal sky, and thus the overall background may not match the QE response pattern measured in the flatfield.

In cases like these, you may want to provide your own "sky" images for use by **pedsky**, rather than rely on the flatfield reference files to represent the shape of the sky. One possibility would be to use sky frames constructed from the median of many dithered science or background exposures. This can sometimes improve the quality of the sky + pedestal fitting even for data taken at shorter wavelengths. This was the approach taken for STScI reductions of the HDF-South NICMOS data, for example. Another possibility, especially for long wavelength data, might be to use a specially-constructed color-dependent flat field (see section 4.4.2). The **pedsky** help pages give further information about this task and its parameters, including guidance for how to use images other than the flatfield as the sky model.

pedsub

As described above, the **pedsky** task requires lots of "blank" sky to be effective, and will only work on relatively sparse NICMOS images. The **pedsub** task provides an alternative method for images which contain larger objects that fill the field of view. The basic methodology for **pedsub** is essentially the same as that of **pedsky**, modeling the image as the sum of a constant (per quadrant) pedestal offset plus an astronomical signal (sky + objects) that is modulated by the flatfield, and then loops over a range of trial values for the pedestal signal, searching for the amplitude which minimizes pixel-to-pixel variations. Unlike **pedsky**, however, **pedsub** can optionally apply a spatial filtering function to each trial image in order to remove unwanted features or spatial frequencies (i.e., the signal from objects) that might bias the calculation of the pixel value spread. The filtering options are "median" and "mask," which essentially carry out low-pass and high-pass filtering of spatial frequencies. The "mask" option removes all large-scale structure from the trial image, leaving the RMS minimization process to operate only on the small-scale, pixel-to-pixel component of the flatfield signal. This option can be effective when trying to measure and remove pedestal from, e.g., images of large galaxies.

Pedsub has many other parameters and options which are fully described in the STSDAS help pages for the task. These should be consulted carefully before using the task, and the user may wish to experiment with combinations of these parameters in order to achieve the best results.

Using nicpipe, biaseq and pedsky: An Example.

Here we illustrate the use of the **nicpipe**, **biaseq** and **pedsky** tasks with one example. A similar sequence could be used with **pedsub** substituted for **pedsky**. The raw data frame is `n4yx23x0q_raw.fits`. This is a NIC3 image taken with the SPARS64 readout sequence and NREAD=24. We might start by using the **sampinfo** task (see section 5.1) to look at the details of the readout sequence.

```
ni> sampinfo n4yx23x0q_raw.fits
```

This will produce a table of output information that looks like this:

```
IMAGE                NEXTEND   SAMP_SEQ    NSAMP       EXPTIME
n4ux23x0q_raw.fits   125       SPARS64     25          1407.933
IMSET                SAMPNUM   SAMPTIME    DELTATIME
1                    24        1407.933    63.997
2                    23        1343.936    63.997
3                    22        1279.939    63.997
...                  ...       ...         ...
21                   4         127.990     63.997
22                   3         63.993      63.388
23                   2         0.605       0.302
24                   1         0.302       0.302
25                   0         0.000       0.000
```

We see that the readouts with linearly spaced DELTATIME values (SAMPNUMs 4 through 24) are in imsets 1 to 21. The last readout (imset 1) often has bias jumps (see “Bias Jumps or Bands” on page 4-10), so we may want to exclude it when feeding the desired range of sky samples to **biaseq** for use in constructing the "clean" sky image. So:

```
ni> nicpipe n4yx23x0q_raw.fits “” stage=biaseq

ni> biaseq n4yx23x0q_ima.fits n4yx23x0q_beq.fits \
>>> skysamps=2-21 fitbias+ fitjump+
```

The output image `n4yx23x0q_beq.fits` has been corrected for non-linear bias drifts and for spatial bias jumps as well.

Next, we complete the pipeline processing using **nicpipe**, doing the FLATCORR, UNITCORR, CRIDCALC steps to prepare for **pedsky**:

```
ni> nicpipe n4yx23x0q_beq.fits “” stage=final
```

We now run **pedsky** non-interactively, letting it fit for the sky level and four quadrant biases on its own.

```
ni> pedsky n4yx23x0q_beq_ima.fits n4yx23x0q_ped.fits \
>>> salgorithm=auto interactive- rmedian-
```

The end product, `n4yx23x0q_ped.fits,` is the fully processed and pedestal corrected image.

Other Pedestal Removal Software

The STScI NICMOS group will continue to explore methods for measuring and removing bias offsets from NICMOS data. In particular, as of the time of this writing, we are investigating the possibility of automatically implementing the **biaseq** and **pedsub** procedures in the **calnica** pipeline. Users should check the NICMOS web pages and STANs for updates on this effort.

In addition, there are other, "freelance" packages for NICMOS data reduction. One example is Brian McLeod's **NICRED** package (McLeod 1997, in the proceedings for the 1997 HST Calibration Workshop, ed. S. Casertano et al., p. 281), which offers a general suite of NICMOS data reduction tools, including routines which estimate and subtract pedestal. Ultimately, there are similarities in the pedestal removal methods used by **pedsky**, **pedsub**, and the **NICRED** algorithms, and therefore it should be possible in principle to unify them in a single task.

4.2 Bars

Bars (figure 4.5) are another bias-related artifact. They appear as narrow stripes that cross the quadrants of an array, and occur identically in all 4 quadrants at the same rows/columns in each. They arise from electrical cross-talk in the detector lines when one camera reads out while the auto-flush pattern is executing in one or both parallel cameras. If an observation was made with all three cameras running in parallel using the same readout sequences, then there usually are no bars present in the data because there is no collision between camera readouts and parallel detector autoflush.

The bars run along the slow readout direction, i.e. along columns in NIC1, and along rows in NIC2 and NIC3. Normally there is only one bar per quadrant, but occasionally there are more, always reflected in all 4 quadrants. Sometimes the bars will be in sync, appearing at the same place in successive readouts. Other times, they may appear in alternate readouts, or their positions may vary from readout to readout, sometimes appearing to march across the frame over the course of a MULTIACCUM sequence. They typically run the length of a quadrant (128 pixels), and are 3 pixels wide. The first pixel is lower than the mean, the second is at the mean level and the third is higher than the mean, giving the impression of an undersampled sinusoidal spike with an amplitude of up to ~10 DN peak-to-peak. The bars are almost always broken in at least one place, with a shift of 2-10 pixels in the narrow direction.

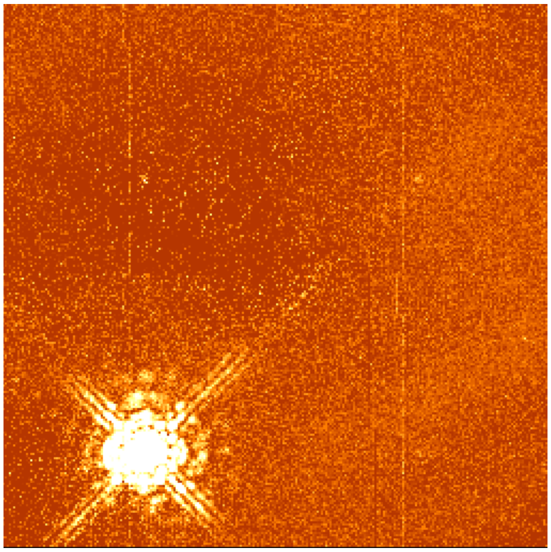
Cures:

The bars are noiseless, localized changes in detector bias. The amplitude is usually quite low, and for some observers they may have little impact on science data. However, it is possible to correct bars during post-processing. One approach is to flag pixels affected by bars in the data quality (DQ) array for each imset where the bars occur, e.g., by setting the DQ values to the bad data flag value 256. Subsequently, when **calnica** carries out the CRIDCALC step, fitting the counts versus time to determine the count rates, it will ignore flagged pixels in a given imset, and the bars will therefore not perturb the fits. A memo describing this procedure is available as a link from the STScI NICMOS Data Anomalies WWW page.

A new pipeline processing step, BARSCORR, is implemented in calnica version 3.3 and higher (see section 3.3). Reprocessing images using this step may eliminate many problems with bars.

If a bar appears in the zeroth readout, it will be subtracted from all the other readouts as part of the normal calibration process (ZOFFCORR), and appear as a negative bar pattern in all readouts of the processed `*_ima.fits` file. This should have no significant effect on the calibrated final image (`*_cal.fits`), however, since the bar just acts as an intercept offset to fit of counts vs. time, which is irrelevant to CRIDCALC which only computes the slope (i.e. the count rate).

Figure 4.5: Bars in a NICMOS camera 1 image.



4.3 Detector Nonlinearity Issues

4.3.1 New Nonlinearity Calibrations

The response of the NICMOS detectors is inherently nonlinear. As was described in section 3.3, the **calnica** pipeline applies a nonlinearity correction with the step NLINCORR. Early nonlinearity calibrations, used for all Cycle 7 and 7N data processed retrieved from the HST Archive before 26 September 2001, used a low-order correction, which worked well for most purposes. However, the STScI NICMOS group has subsequently recalibrated the detector nonlinearities and produced improved NLINFILE reference files. There are two significant changes:

- In the old nonlinearity calibrations, data values below a minimum threshold (set for each pixel by the value in the `[NODE,1]` extension of the nonlinearity reference file) had no calibration applied. However, NICMOS data are actually nonlinear at all levels, and in the new reference files the linearity correction will be calibrated and applied at all count levels.

- Previously, only corrections of the form $F_c = (c_1 + c_2*F) * F$ were allowed. The new reference files and software will permit 2nd order corrections terms which improve the accuracy of the linearity calibration (see section 3.3).

The new nonlinearity reference files are now used by default when data are retrieved from the archive and processed by OTFR. Version 3.3 of **calnica** accommodates second order correction scheme used in the new reference files.

4.3.2 Non-Zero zeroth Read Correction for Bright Sources

As described in section 4.1.2 above, the first non-destructive readout after a reset provides the reference bias level for the counts in each pixel of a NICMOS science image. This "zeroth readout" is transmitted to the ground with MULTIACCUM data, but is directly subtracted from the final readouts on-board the telescope for all the other readout modes. Due to physical limitation in the readout speed, the zeroth read happens 0.203 seconds after the reset of the detector. NICMOS does not have a shutter, and therefore when a bright source is being observed, a non-negligible amount of charge may accumulate on the detector by the time the zeroth read is performed, enough to significantly affect detector nonlinearity. However, zeroth read subtraction is the first step of the calibration processing for MULTIACCUM data (ZOFFCORR in **calnica**), and is automatically done on-board the telescope from the final read in a ACCUM exposure. The zeroth read signal information is therefore lost before nonlinearity correction (NLINCORR in **calnica**) is performed, and for bright sources this correction may therefore be inaccurate.

Cures:

In November 1997, an additional step was added to the **calnica** pipeline processing software (version 3.0 and higher) to account for this zero read signal correction, and an additional keyword, ZSIGCORR, was included in NICMOS image headers to control this step. The ZSIGCORR step computes an estimate of the number of counts for each pixel in the zeroth readout of a MULTIACCUM, based on the count rate measured later in the exposure. This information is then used in the NLINCORR step to estimate the absolute charge in each pixel and the appropriate linearity correction for that charge level. The optional **calnica** parameter zsthresh controls a threshold above which the ZSIGCORR correction is made.

If you have observed very bright targets, you should read the calnica documentation concerning ZSIGCORR.

Pipeline-processed data that were retrieved from the STScI Archive before November 1997 has no ZSIGCORR correction. When reprocessing such data, the current version of **calnica** will automatically apply this step to all MULTIACCUM mode observations if both ZOFFCORR and NLINCORR are also being performed.

4.3.3 Uncorrected Saturation

Occasionally, for reasons which are not fully understood, NICMOS MULTIACCUM images of bright targets will saturate at an ADU level which is below the nominal saturation threshold specified in the nonlinearity calibration reference file. This could happen, for example, when the variable quadrant bias levels (pedestal) are at unusually large values. When this happens, the accumulating counts vs. time for the affected pixels level out at the saturated value. However, because the saturation flag is not set at the NLINCORR stage of **calnica**, when the CRIDCALC step fits "up the ramp" of counts vs. time to determine the count rate slope, it does not know to exclude the saturated data values, and fits a linear slope to a highly nonlinear data set. Among other unpleasant consequences, this tends to trigger the CRIDCALC cosmic ray detection flags, causing wholesale rejection of many or most time steps for the affected pixel. Images which have been affected by this problem will generally have anomalously low count rates for bright pixels, but the saturation flag (64) will not be set in the DQ arrays.

Cures:

The simplest cure is to modify the nonlinearity reference file to reduce the threshold where saturation flagging is triggered, and then to reprocess the data. The saturation level for each pixel is set in the [NODE,2] extension of the nonlinearity reference file. As an example, let us say that the science image is called n4ux22hbq_raw.fits and the corresponding nonlinearity reference file is h7n1654cn_lin.fits. We reduce the saturation threshold by 5%, edit the science image primary header to point to the new reference file, and reprocess:

```
ni> copy h7n1654cn_lin.fits new_nonlin.fits
ni> imarith new_nonlin.fits[NODE,2] * 0.95 \
>>> new_nonlin.fits[NODE,2][overwrite+]
ni> hedit n4ux22hbq_raw.fits[0] NLINFILE new_nonlin.fits \
>>> update+ verify- show+
ni> calnica n4ux22hbq ""
```

If the problem reoccurs, reduce the [NODE,2] values further until the saturation threshold is properly triggered.

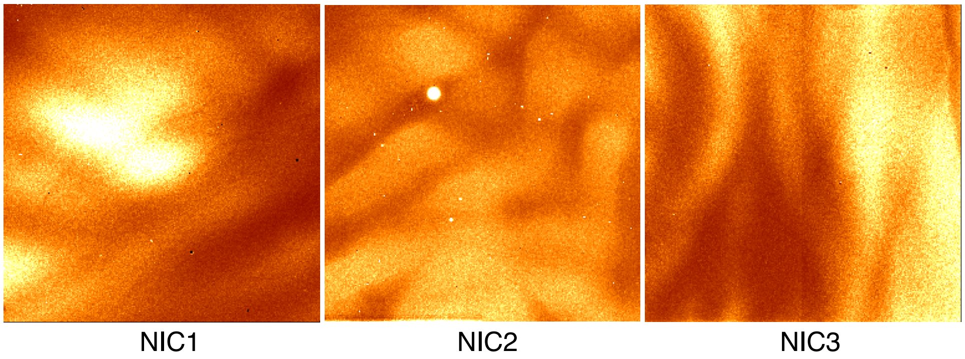
4.4 Flatfielding

4.4.1 Characteristics of NICMOS Flatfields

NICMOS flatfields show significant large-scale non-uniformity as well as pixel-to-pixel fluctuations (figure 4.6). The flat field variations are also a strong function of the wavelength. At 0.8 µm, there is variation by a factor of ~5 (minimum to maximum) in the relative response across the array. This declines to a factor of ~3 at a wavelength of 2.2 µm, and at 2.5 µm the array is almost flat. Naturally, both the spatial distribution and amplitudes of these variations are different for each camera. Camera 3 has an overall side-to-side response gradient, making one half of the array generally more efficient than the other. Camera 1 has a large, contiguous region with low sensitivity.

An inevitable consequence of these large QE variations is that science data will have large spatial variations in their pixel-to-pixel noise level, and hence in signal-to-noise for sources. For images where statistics are limited by photon shot noise (either from the sources themselves or from the sky background) the S/N variations scale as the square root of the flat field amplitude (e.g., a factor of ~2 at F160W). Many NICMOS observations, however, especially with Cameras 1 and 2, are largely readout noise limited and for these the signal-to-noise modulation will scale linearly with the flat field, e.g., a factor of ~4 at F160W.

Figure 4.6: NICMOS flatfields. From left to right, NIC1 F190N, NIC2 F237M, and NIC3 F222M. Lighter shading indicates regions of lower sensitivity. The coronagraphic hole is visible in the NIC2 image.



4.4.2 Temperature-dependent Flatfields

As with the dark current and bias, the quantum efficiency of NICMOS detectors is a function of the detector temperature. Because the instrument temperature varied somewhat throughout Cycle 7, the flat field structure also varied slightly due to this effect. For the temperature range experienced during Cycle 7, this was a relatively small effect (a few percent), but careful users may wish to recalibrate their data using flat fields customized for the operating temperature of the instrument at the time of their observations. The process here is essentially identical to that described in section 4.1.5 for temperature dependent dark correction. The STScI NICMOS group has created a WWW-based tool that will generate temperature-specific flat fields appropriate for any NICMOS camera and filter over a range of detector temperatures. This capability may become particularly important in Cycle 11 and afterward, when the NICMOS Cooling System (NCS) is expected to result in instrument operations at a substantially higher temperature than was used in Cycle 7.

Detector temperatures are stored in the NDWTMP11 and NDWTMP13 keywords in the *_spt.fits files. For Cameras 1 and 2, use NDWTMP11; for Camera 3, use NDWTMP13.

4.4.3 Color Dependence of Flatfields

The strong wavelength dependence of the NICMOS response may affect the quality of flatfielding for objects with extreme colors. The color of the internal flatfield lamps may not match that of the sky background, or indeed of many astronomical sources. In most cases this should only affect photometry by a few percent, but concerned users may wish to experiment with constructing customized, spectrally weighted flatfields. These can be constructed from linear combinations of (and interpolations between) existing NICMOS narrow band flat fields, as described in Storrs, Bergeron and Holfeltz 1999 (ISR NICMOS 99-002).

One situation where the color dependence of the flat fields may have a visibly noticeable effect is on the flatfielding of the sky background. The background has a significantly different spectrum than the flat field lamps, particularly at $\lambda > 1.8\mu m$, where thermal emission from the telescope and instrument become important. After dividing by ordinary internal lamp flats, flat field residuals may still be present in the sky background. Unfortunately, in practice it is difficult to separate these from the characteristic variations induced by the pedestal effect discussed above in section 4.1.2. Pedestal removal routines like **pedsky** (see section 4.1.5) key off these residual flat field variations, and depend on having an accurate model of the spatial structure of the sky. If actual, observed sky frames are not available, then synthetic color-dependent flats may be useful when running routines like **pedsky**.

Cures:

A discussion of flat field color dependence is presented in Storrs, Bergeron and Holfeltz 1999 (ISR NICMOS 99-002). That paper describes methods for constructing color-dependent flatfield reference files, and provides IRAF scripts for doing so.

In general when using NICMOS data, you should keep in mind the large spatial variations in sensitivity and signal to noise that result from the flat field structure. This can have important consequences when you are analyzing NICMOS data, requiring, e.g., careful handling when setting source detection thresholds in automatic routines like **DAOPHOT** or **SExtractor**, or for error analysis on photometry.

4.5 Pixel Defects and Bad Imaging Regions

4.5.1 Hot Pixels, Cold Pixels, and Grot

Each NICMOS array has a small number of bad pixels which may be either *cold* (i.e., very low or zero response) or *hot* (i.e., with very high or erratic dark current). Most of these were mapped during thermal vacuum testing, and static bad pixel masks are used to set error flags in the ERR arrays during the MASKCORR step of **calnica** processing. The statistics on the cold and hot pixels present in each of the NICMOS cameras are presented in table 4.1.

In addition to the bad pixels which were already known from ground-based testing, more pixels have shown low measured quantum efficiency in orbit. Now known as *grot*, they appear as small areas of reduced sensitivity, most likely due to flecks of anti-reflective paint which were scraped off the optical baffles between the dewars as they were forced against each other by the solid nitrogen cryogen expansion. The largest example, known as the *battleship*, is found in camera 1 and affects approximately 35 pixels. However, most grot seems to affect single pixels. Scrape tests conducted at Ball Aerospace shortly after NICMOS was launched reveal that the paint flecks can range in size from 25μm to greater than 100μm. Since NICMOS pixels are 40μm on a side, this means there is potential for grot smaller than 1 pixel.

Table 4.1: NICMOS Cosmetics

Type	NIC 1	NIC 2	NIC 3
Hot Pixels	68	94	17
Cold Pixels	10	11	3
Grot	284	227	288

Cures:

Dithering is the best way to guard against the effects of bad pixels and grot. If you have multiple dither positions, you can use the static masks and the grot masks to exclude bad pixels when combining the images (e.g., with **calnicb**, or with IRAF **imcombine**, or with the STSDAS **dither.drizzle** routines). Or, you may interpolate over masked pixels using, e.g., IRAF **fixpix**. Grot masks are available on the NICMOS instrument website and should be used to correct data taken between June 1997 and November 1998. For further description of these files, see Sosey & Bergeron 1999 (ISR-NICMOS-99-008). The grot masks are *not* presently used as part of the default calibration procedure and are only presented to warn users about the position of problem pixels. Although there is a DQ flag value (16) reserved for grot (see table 2.2), the default MASKFILE reference images used by **calnica** do not include grot. It would be straightforward, however, to make your own bad pixel mask, combining the information about hot and cold pixels found in the current MASK reference file with the locations of pixels identified in the grot mask. You may then reprocess from scratch using this new MASKFILE, or you may add the new grot information into the DQ array of an already reduced data set. When the dithered images are then combined by **calnicb**, the flagged pixels will automatically be excluded.

Not all pixels affected by grot are unrecoverably ruined. For pixels which are only partially obscured, the effects of grot may flatfield out. You may wish to compare your flatfielded images with the grot masks to see which pixels have flattened well and which are potentially suspect.

4.5.2 Erratic Middle Column/Row.

In each NICMOS camera, there is one row or column which often deviates from others nearby. In NIC1 this is row 128; in NIC2 it is column 128, and in NIC3 it is column 129. Adjacent rows (or columns) are sometimes affected to a lesser degree. It is believed that this "photometrically challenged column" may be related to uncertainties in detector shading corrections. The affected column/row contains the first pixels read out in each quadrant. Since the shading function is very steep and highly nonlinear during this part of the readout, it is the part of the array most sensitive to changes in the detector environment. Thus it is thought that the pixels in these rows or columns are not any less sensitive, but that they have just had an incorrect bias subtracted from them. The result is a row or column of pixels that is either under or over corrected for the shading by the dark reference file.

Cures:

For some data, it is possible to fit a function to the affected column and add it back in as a delta bias, but the data in the affected pixels tends to be rather noisy as well. Given dithered data, you may be better off just treating the affected pixels as "bad" and using spatial dithers to recover that information (just as you would do for other bad pixels). As with grot (section 4.5.1), adding the middle column/row to the DQ array flags of the calibrated image (`*_cal.fits`), or to the MASKFILE before **calnica** processing, will mark them for exclusion when dithered images are subsequently combined by **calnicb**.

4.5.3 Coronagraphic Hole

NICMOS camera 2 includes a coronagraphic hole which can be used to partially obscure bright sources. In non-coronagraphic observations, however, the hole is still present, and will partially or completely blank out any objects which fall near it. For thermal IR images, the hole can appear as a positive "bump" due to excess background emission shining through the hole from warmer structures behind it. The position of the hole in detector coordinates drifted with time during the lifetime of the instrument, changing rapidly shortly after launch when the cryogen distortion of the dewar was varying quickly, then slowly stabilizing.

Cures:

The hole should usually be treated as "bad pixels" when combining dithered NIC2 images, either by masking or interpolation. Masking a circular region with radius ~7 pixels should eliminate the effects of the hole for most data sets. Tables and graphics showing the location of the coronagraphic hole as a function of time are available on the STScI NICMOS Instrument WWW page at:

http://hst.stsci.edu/nicmos/performance/coronagraphy.

You may also retrieve information about the coronagraphic hole position for a given dataset using the NICMOS History Tool (see section 5.2).

For coronagraphic imaging science, see the discussion on reducing coronagraphic data in chapter 5.

4.5.4 Vignetting

There are two effects which are sometimes described as *vignetting* in the NICMOS cameras.

First, the anomalous expansion of the NICMOS dewar moved the cameras in such a way that they image the edge of the Field Divider Assembly (FDA) at the bottom of all 3 cameras. The FDA is a fold mirror that sends the light from the foreoptics down each camera channel and through the filters and dewar window to the cameras. The bottom edge of the field of view for all 3 cameras looks at a black mask on the FDA. This means that there is less signal from the telescope along that edge, and thus the throughput is decreased. In NIC3 ~15 rows along the bottom edge are severely affected, and the throughput there ramps down steeply to about 30% of the mean for the rest of the array. For NIC2 it has about the same extent, but is only reduced to 95% of the mean of the rest of the array. The effect for NIC1 is even smaller. This hard edge moved around slightly with time, and thus does not flatfield away perfectly, especially for camera 3 where the gradient is steep. Because the vignetting is small for cameras 1 and 2, the resulting flatfielding uncertainty is generally negligible. (The coronagraphic hole in camera 2 is also on the FDA, so its motions correlate with those of the vignetted region.)

In addition, the NICMOS cameras also imaged part of the forward bulkhead at the entrance aperture. This warm metal surface emits radiation in the longer wavelength bands. This emission extends over about 50 rows in NIC3 and about 20 rows in NIC2. Moving the Field Offset Mechanism (FOM) causes NICMOS to see a different part of the sky in the HST focal plane, and essentially to look farther away from the bulkhead that causes the emission. This FOM repositioning was automatically implemented for all camera 3 observations from January 1998 (the first NIC3 refocus campaign) onward, so data taken after that time are unaffected. Data in the longer wavelength filters taken prior to that time may have an elevated gradient in the sky background. The FOM is forward of the FDA in the optical path, so the repositioning has no effect on the FDA vignetting described above.

Another consequence of the vignetting is its impact on image shape and geometric distortion for NIC3 images taken out of focus, i.e. at any time outside the two NIC3 refocus campaigns (January and June 1998). The vignetting introduces some additional image distortion, especially over the first ~50 rows of the detector, when the camera is out of focus (see, e.g., Cox et al. Instrument Science Report OSG-CAL-97-07). Users should regard astrometry and PSF shape in that portion of non-campaign NIC3 images with some caution.

Cures:

For camera 3 data, it may be prudent to discard data from the bottom 10 to 15 rows. At the very least, photometry in that region should be treated with suspicion. Data taken prior to the FOM repositioning in January 1998 may have elevated background gradients for camera 3.

For NIC3 images taken outside of the refocus campaigns, image shape and astrometry over the bottom ~50 rows should be regarded with caution.

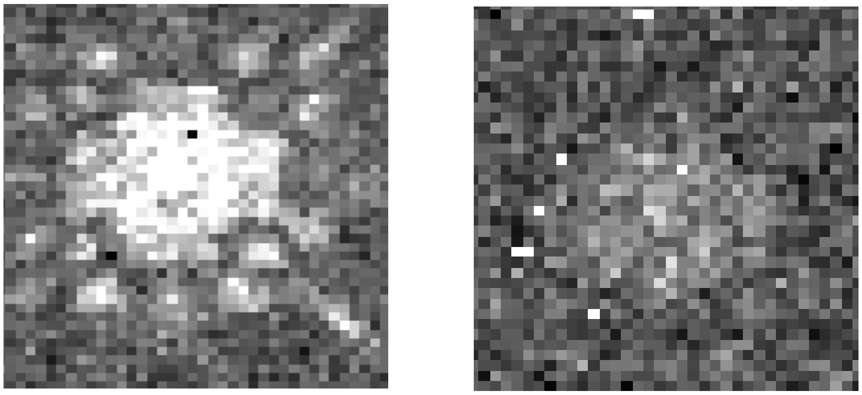
4.6 Effects of Overexposure

Because each pixel of the NICMOS detectors is read individually, overexposure does not cause "bleeding" along columns or rows, as is seen with CCDs. Exposure to bright sources, however, can result in two other sorts of NICMOS artifacts: persistent images, and electronic ghosts known colloquially as the "Mr. Staypuft" anomaly.

4.6.1 Photon Persistence

HgCdTe detector arrays are subject to image *persistence*. When pixels collect a large amount of charge, they will tend to "glow" for some time after the end of the exposure. This persistent signal decays exponentially with a time scale of about 160 ± 60 seconds, but there is also a long, roughly linear tail to the decay such that persistence from very bright sources remains detectable as much as 30 to 40 minutes after the initial exposure. Exposures of bright astronomical targets therefore can leave afterimages which appear in subsequent exposures taken within the same orbit, or even into the next (figure 4.7). If you are analyzing dithered exposures of a bright target, you may wish to carefully inspect each image for residual ghosts from the previous exposure(s), and perhaps mask them out when coadding the dithered images. It appears that all sources of illumination probably leave persistent afterimages, but under typical conditions they are most noticeable for sources which have collected 20,000 or more ADU during the previous exposure. It is not unreasonable to expect afterimage signals of ~1 e^-/second immediately following a severe over-exposure. Occasionally, persistent images result from bright targets imaged by the *previous* NICMOS observer.

Figure 4.7: Persistence induced by an exposure of a bright star. The left and right images show dark frames taken 32 and 512 seconds after the exposure of the star.



32 seconds delay

512 seconds delay

Cures:

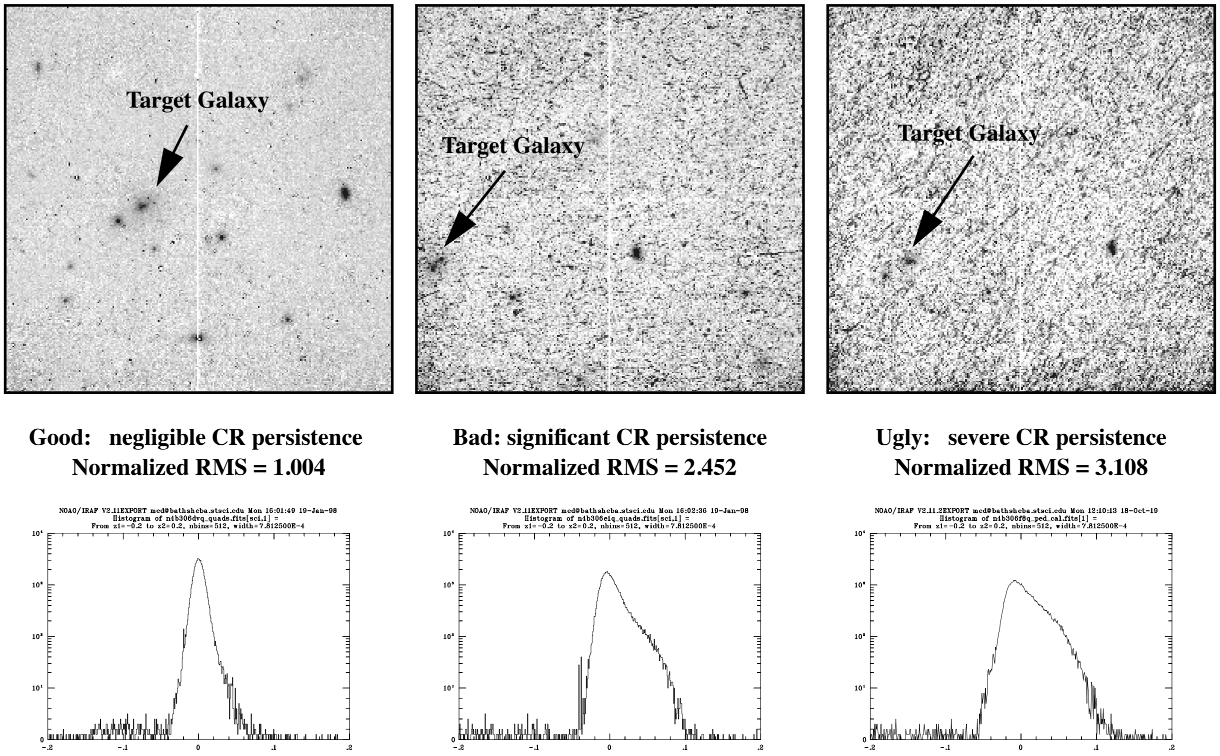
There is no easy cure for persistence in existing data; just be aware that it may be present. For dithered observations you can mask out persistent images from previous exposures before combining into a mosaic. It may be possible to subtract a rescaled version of the previous image that *caused* the persistent afterglow from the persistence image itself and thus remove it (see the discussion of cures for cosmic ray induced persistence below)

4.6.2 Cosmic Ray Persistence

More insidiously, during regular passages of HST through the South Atlantic Anomaly (SAA), the arrays are bombarded with cosmic rays, which deposit a large signal into nearly every pixel. The persistent signal from these cosmic rays may then be present as a residual pattern during exposures taken after the SAA passage. This appears as a mottled, blotchy or streaky pattern of "noise" (really signal) across the images, something like a large number of faint, unremoved cosmic rays. These persistent features cannot be removed by the MULTIACCUM cosmic ray processing done by the standard pipeline (i.e. the CRIDCALC step of **calnica**) because they are not transient. Rather, they are a kind of signal, like a slowly decaying, highly structured dark current.

Cosmic ray persistence adds non-Gaussian, spatially correlated noise to images, and can significantly degrade the quality of NICMOS data, especially for exposures taken less than 30 minutes after an SAA passage (see examples in figure 4.8). Count rates from moderately bad cosmic ray persistence can be of order 0.05 ADU/second, with large pixel-to-pixel variations reflecting the spatial structure of the signal. The effective background noise level of an image can be increased by as much as a factor of three in the worst cases, although 10% to 100% is more typical. This "noise" is primarily due to the spatially mottled structure in the persistence, not the added Poisson noise of the persistence signal itself. Because HST passes through the SAA seven or eight times a day, a large fraction of NICMOS images are affected by cosmic ray persistence to one degree or another. Observations of bright objects are hardly affected, since the persistent signal is usually quite faint. Similarly, short exposures are not likely to be badly affected because the count rate from persistence is low and may not exceed the detector readout noise. But deep imaging observations of faint targets can be seriously degraded. The NICMOS Instrument Science Report ISR-98-001 (Najita, Dickinson and Holfeltz 1998) presents a detailed discussion of this phenomenon and its effects on imaging observations.

Figure 4.8: Cosmic ray persistence in three dithered NIC2 MIF1024 images. For each image, the histogram at bottom shows the distribution of pixel values near the sky level. The shoulder of positive values for the persistence-impacted images demonstrates the non-Gaussian nature of CR persistence noise. The "normalized RMS" is the measured value divided by the mean for many "clean" images free of CR persistence.



Good: negligible CR persistence
Normalized RMS = 1.004

Bad: significant CR persistence
Normalized RMS = 2.452

Ugly: severe CR persistence
Normalized RMS = 3.108

Cures:

If you have multiple exposures within a given SAA-impacted orbit, the amplitude of the persistence should decay, and you may want to give later exposures higher weight when combining them. Sometimes it may be preferable to discard the worst frames altogether. Well-dithered data lessens the impact of persistence, since objects will move relative to the persistent CR signal and it will not sum coherently when the data are registered. With well-dithered data (at least three positions), one can also take advantage of the **drizzling** procedure and associated software in the STSDAS **dither** package to identify and mask the worst effects of persistence, as described, e.g., in the "Drizzling Cookbook," (Gonzaga et al. 1998, STScI ISR WFPC2 98-04).

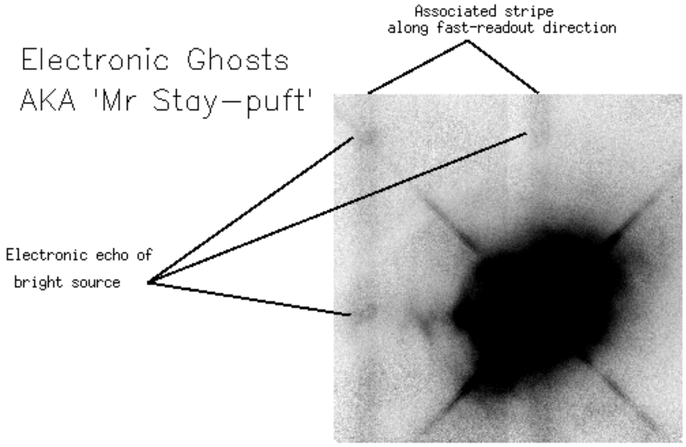
A web-based tool for determining the time since SAA passage for any given data set is now available from the STScI NICMOS web pages. The tool is also linked to the NICMOS History Tool (see section 5.2). It accepts a list of observation times (specified as modified Julian dates, MJD - these can be obtained from the EXPSTART image header keyword), and generates a list of the elapsed time and durations of the preceding SAA passages, as well as a graphical display showing the observation and SAA times. Because the amplitude of cosmic ray persistence in NICMOS images depends strongly on both the time elapsed since SAA passage and on the duration of the passage itself, this tool provides a useful way of quickly "pre-screening" your data to see which frames are most likely to be contaminated.

On rare occasions, it may happen that dark exposures were taken after an SAA passage but before science observations. These darks can provide a "map" of the cosmic ray persistence, and in principle it is possible to scale and subtract them from the subsequent science exposures in order to minimize the effects of the cosmic ray afterglow. This has even been done using short science exposures taken early in the orbit, scaled and subtracted (after masking out objects) from the science images that follow. If you have multiple exposures in an orbit, you may wish to consider trying this, but at present there is no general software available for doing this procedure. Because the CR persistence pattern remains fixed in successive exposures, you may also use the first exposure in the orbit to create a mask for setting the affected pixels to zero weight when combining the images. In HST Cycle 11, when NICMOS is revived, STScI will automatically schedule special dark exposures after SAA passages in order to provide a map of the CR persistence. Software for implementing this persistence correction will be written and tested as soon as possible after SMOV3B, when the NCS is installed.

4.6.3 Amplifier Ringing (The "Mr. Staypuft" Anomaly)

There is another sort of NICMOS ghost image, wholly separate from those induced by persistence. Bright targets which appear in a given detector quadrant can produce an electronic *ringing* effect in the readout amplifiers, which may induce ghost images at the same pixel locations in the other three quadrants. This is believed to be due to the pull-down of the power supply, which does not completely recover from reading a large number by the time it's asked to read the next number (from the next quadrant). In some cases a whole row (or column, whichever direction the "fast" read clocking runs) can be anomalously high. The amplitude of the ghost images is of order a tenth of a percent of the real image count rate. Inside the STScI NICMOS group, this effect is fondly known as the "Mr. Staypuft anomaly" (figure 4.9).

Figure 4.9: The Mr. Staypuft anomaly. The bright source at lower right produces electronic ghost images in the other three quadrants, plus vertical streaks. In Camera 1 images, these streaks run horizontally.


Associated stripe along fast-readout direction
Electronic Ghosts AKA 'Mr Stay-puft'
Electronic echo of bright source

Cures:

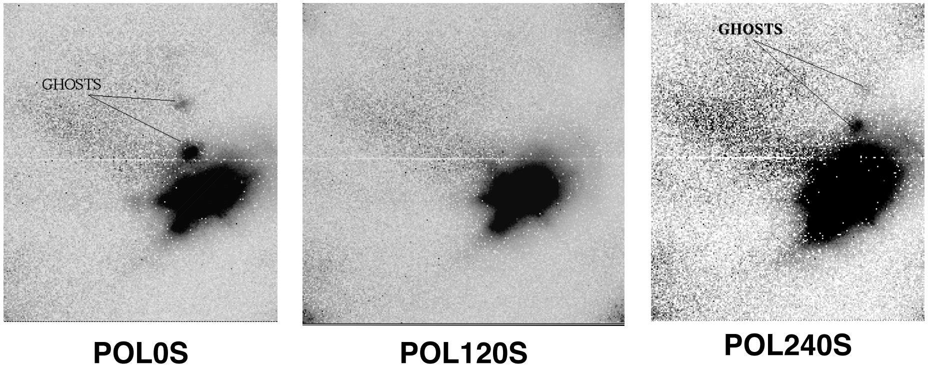
At present there is no real solution to the Mr. Staypuft anomaly other than to be aware that such ghost images may exist. For example, if your image has a bright source at pixel (58,143), then you may see ghost images around (186,143), (186,15), and (58,15). If the observation was made with Camera 3 (so that the fast clocking direction is along columns), there may also be elevated signal at all rows on and around columns 58 and 186. Ordinarily it is not possible to eliminate these ghosts from dithered data by masking, since they will move about on the array along with the astronomical targets. If, for some reason, observations were obtained at multiple roll angles, then it would be possible to mask the ghosts and eliminate them; this was done for the Hubble Deep Field South NICMOS observations.

Some NICMOS observers studying relatively blank fields have dealt with the "elevated columns/rows" phenomenon by fitting medians to the affected columns/rows (or to portions of the columns/rows, avoiding regions affected by bright or extended targets), and subtracting the median values from each column or row of the data. This can at least provide a cosmetic fix. In principle, this correction should probably be applied *before* flatfielding the images, since it is most likely that the "Staypuft" signal is not responding to the array QE variations, although this has not yet been formally tested. However, if the fit is done before flatfielding, then the sky background itself will not be uniform, and in fact will be improperly subtracted by the procedure. For this reason, in practice it is probably safer (if not strictly correct) to apply a median column (or row, for NIC1) correction *after* flatfielding instead. The STScI NICMOS group is now experimenting with algorithms to correct the Staypuft column/row effect in post-processing, based on an empirical model for its electronic behavior. A software tool for correcting data may be released sometime in the future, and it is possible that the procedure may be automatically implemented as a step in the **calnica** pipeline.

4.6.4 Optical Ghost Images

In addition to persistence and electronic ghosts, NICMOS polarimetric images of bright targets are subject to optical ghosts (figure 4.10). These are most evident for the short wavelength polarizers in NIC1. The location of the polarizer ghosts relative to the source changes direction with respect to the principal axes of transmission. For POL0S, the brightest ghost is found roughly (-10,+35) pixels away from the target position, and a second one at (-16,+70) pixels. It is possible that for a brighter object more ghosts would appear in the same angle and direction. POL120S images appear to be ghost-free. For POL240S, ghosts appear (+10,+54) pixels from the source position and (+18,+80) pixels, with the possibility of more appearances for brighter targets.

Figure 4.10: Optical ghosts in NIC1 polarizers.



Ordinary (non-polarimetric) NICMOS images of bright sources may have very faint filter ghosts which result from back-reflections off the faceplate and the filters. The position of the ghosts changes from one filter to another. They are roughly 8 magnitudes fainter than the star which produces them. The ghosts cannot be completely subtracted from the field unless the reference PSF is at the exact same location as the star. They are typically seen as a residual in the PSF subtracted images, and look like large "donuts" because they are out-of-focus reflected images.

Cures:

No true cure is possible: be aware that such ghosts may exist; their positions are predictable. If images are available with multiple spacecraft roll angles, the ghosts may be masked out before combining the data.

4.7 Cosmic Rays of Unusual Size

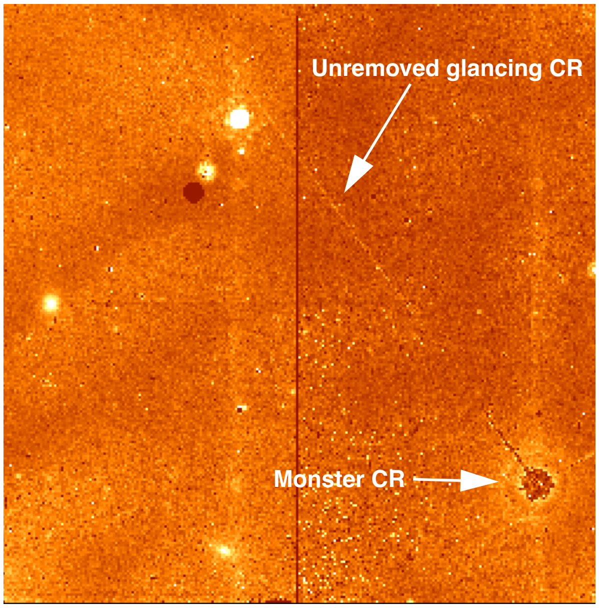
As with CCDs, cosmic ray (CR) hits produce unwanted signal in the output images. Unlike standard CCD observations, however, most of the effects of CRs in NICMOS data can be eliminated during data processing thanks to the capability for multiple, non-destructive readouts in infrared arrays. As was described in chapter 3, the **calnica** processing pipeline identifies cosmic rays in individual readouts of a MULTIACCUM image during the CRIDCALC step, and excludes data from that readout when calculating the final count rate for the affected pixel. Therefore CRIDCALC processed NICMOS images (`*_cal.fits`) should be free of most or all ordinary cosmic rays. The affected pixels will be flagged in the DQ extensions of the `*_ima.fits` files.

It is worth noting that the noise level will be slightly higher for pixels which were impacted by cosmic rays because (1) the effective exposure time for those pixels is shorter because one readout has been discarded, (2) breaking the counts vs. time "ramp" fitting procedure into two or more pieces introduces extra statistical noise, particularly for observations limited primarily by readout noise, and (3) although the cosmic ray signal itself is removed, the associated Poisson noise is not.

Images taken in ACCUM mode have no cosmic ray processing. For these, you must handle cosmic rays as you would for ordinary CCD images.

Occasionally NICMOS (like other instruments) will be hit by a very strong cosmic ray, that can affect many pixels (figure 4.11). These will not always be effectively removed by the **calnica** CRIDCALC processing. The core of the very bright CR will be flagged and removed, but there can be a large, surrounding "halo" of pixels weakly affected by CR signal that are not flagged. Also, glancing-incidence cosmic rays can leave long trails across an image, and the weaker CR pixels in the streak may not be flagged by the CRIDCALC processing. Very strong CRs can induce persistence which will alter the count rate for the rest of the MULTIACCUM sequence, resulting in a cosmic ray afterimage. If pixels reach saturation levels from a cosmic ray impact, subsequent readouts are unrecoverable.

Figure 4.11: A "monster cosmic ray" hit in a NICMOS image. The brightest pixels of the CR have been cleaned by CRIDCALC in **calnica**, but a surrounding halo of affected pixels remains, as does a long diagonal streak of unflagged pixels from the CR's glancing traversal of the array. Also note the vertical "Mr. Staypuft" streaks (see section 4.6.3) in both the left and right halves of the image, which are evidently induced by the CR impact itself.



Cures:

If weakly affected pixels around bright cosmic rays are not being flagged by CRIDCALC, you may wish to reprocess the data, reducing **calnica**'s CR rejection threshold parameter *crthresh* from the default (4) to a smaller value. Alternatively, you may flag the cosmic ray affected pixels manually before CRIDCALC processing. First, partially process the images through **calnica** excluding the CRIDCALC stage (i.e. setting CRIDCALC to OMIT). Next, in the resulting `*_ima.fits` image, edit the DQ extension of the first IMSET affected by the cosmic ray, setting the affected pixels to the cosmic ray flag value 512 (see table 2.2). Finally, set CRIDCALC to PERFORM in the `*_ima.fits` file and complete the **calnica** processing using the modified `*_ima.fits` as input.

The technique described here for flagging monster cosmic rays can also be used to flag and eliminate trails from moving targets (usually space junk) which occasionally cross through NICMOS images.

4.8 Scattered Earthlight

All HST instruments are occasionally subject to scattered earthlight which enters when the telescope is pointing near the bright earth limb. This results in an elevated background level, which does not necessarily illuminate the detector in the same way as the sky does, and thus does not flatfield away properly. For NICMOS MULTIACCUM observations, scattered light during a portion of the exposure (usually either the beginning or the end) will also cause the signal level to accumulate in a non-linear way with time, i.e. either with an initial or final "ramp-up" where the scattered light elevates the count rate for some of the readouts. This can cause problems when the **calnica** CRIDCALC routine fits a linear function to counts vs. time, triggering the rejection algorithm for some pixels and not others, and resulting in erroneous slope fits (and hence, incorrect derived count rates) for some pixels.

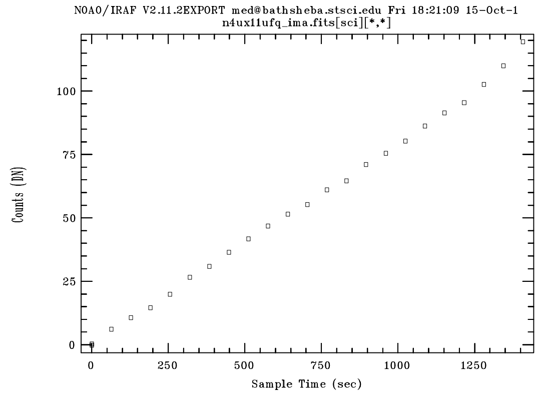
Cures:

Fortunately, for a MULTIACCUM observation, one can take advantage of the time resolved nature of the data to exclude individual readouts where scattered light contaminates the data. As an example, let us look at one data set where scattered light plays a role. In this example, we will also use some of the tools in the **stsdas.hst_calib.nicmos** package which are described in more detail in chapter 5. First, we will partially process the image through **calnica** using **nicpipe**, skipping the UNITCORR, FLATCORR and CRIDCALC steps. The resulting `*_ima.fits` image is thus in units of counts, not countrate. Next, we use the **sampdiff** task to plot the median counts in each IMSET of the image vs. the sample time.

```
ni> nicpipe n4ux11ufq_raw.fits ““ stage=biaseq

ni> pstats n4ux11ufq_ima.fits \
>>> extname=sci units=counts stat=midpt
```

The result is shown in figure 4.12:

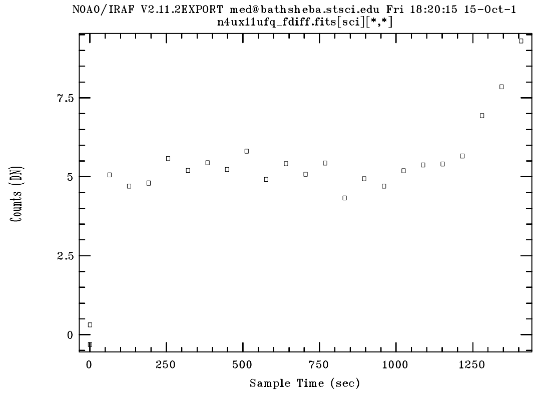
Figure 4.12: Plot of median counts vs. time for readouts from a NICMOS image. The upturn in the last few readouts shows the effects of scattered light.



The upturn in the counts during the last few readouts indicates the presence of scattered light at the end of the exposure. It is sometimes easier to see the effects of scattered light if you look at the "first differences" in an image, i.e., the *difference* between each readout and the preceding one. In that way, you see only the counts that were accumulated during that readout sample, and not the cumulative sum of everything that came before. This can be done using the task **sampdiff** in the **nicmos** package (again, see chapter 5).

```
ni> sampdiff n4ux11ufq_ima.fits n4ux11ufq_fdiff.fits

ni> pstats n4ux11ufq_fdiff.fits \
>>> extname=sci units=counts stat=midpt
```

Figure 4.13: Median counts per IMSET for the "first difference" image computed from the *_ima.fits file using sampdiff. The upturn due to scattered light is more prominent when viewed this way.



The easiest and safest way to eliminate the effects of scattered light is to discard the affected readouts before fitting with CRIDCALC to compute the count rates and identify the cosmic rays. This will cost you part of the exposure time in your image, but will eliminate the elevated background, the uneven illumination, and problems resulting from incorrectly fit count rates and accidental triggering of the CRIDCALC cosmic ray rejection. This can easily be done by editing the data quality arrays in the DQ extensions of the affected images, setting them to some non-zero value (4096 is a good choice, as it is not otherwise used as a DQ flag value for NICMOS). This will cause CRIDCALC not to use those imsets when fitting a slope to counts vs. time to derive the count rate. You can edit these in the raw data frame and start over with the processing, or (in the example we are using here) in the partially processed frame, and then complete the processing using **nicpipe**. For our example, we might continue in the following way. We want to exclude the last three readouts from further processing. Remember that NICMOS imsets are stored in reverse order in the multiextension FITS file, i.e., [SCI,1] is the last readout, [SCI,2] the next to last, etc.

```
ni> copy n4ux11ufq_ima.fits n4ux11ufq_imaflag.fits
ni> imreplace n4ux11ufq_imaflag.fits[DQ,1] 4096 ver- up+
ni> imreplace n4ux11ufq_imaflag.fits[DQ,2] 4096 ver- up+
ni> imreplace n4ux11ufq_imaflag.fits[DQ,3] 4096 ver- up+
ni> nicpipe n4ux11ufq_imaflag.fits "" stage=final
```

The end product will be called `n4ux11ufq_imaflag_cal.fits`, and should be free of scattered light, albeit with a somewhat shorter effective exposure time. The actual exposure time used per pixel can be determined by looking at the `[TIME,1]` array of the final, calibrated image.

CHAPTER 5:

Data Analysis

In this chapter. . .

This chapter describes specific tools and topics related to the analysis of NICMOS data. The first section points out some STSDAS tools for analyzing NICMOS images. The tools described here are specialized for use with NICMOS, and supplement the more general STSDAS tasks described in chapter 3 of the HST Introduction. The remainder of the chapter deals with topics of more or less general interest: photometric calibration, PSF subtraction, coronagraphic reductions, polarimetric analysis and grism data reduction.

5.1 STSDAS Software

Software tools for NICMOS FITS files available in the STSDAS packages **toolbox.imgtools.mstools** and **hst_calib.nicmos** have been designed to maintain compatibility with pre-existing analysis software. The tools have either been written in ANSI-C or are IRAF CL scripts interfacing with pre-existing IRAF/STSDAS tasks. Some of the new tools accept a variety of data formats such as OIF and GEIS, as well as STIS and

NICMOS FITS files, and will in time replace the STSDAS tasks they render obsolete.

The new tasks fall into two major categories:

1. ***General-purpose utilities***. These tasks include tools for mathematical and statistical operations on science images and for analysis and display of reduced and raw data. In most cases, the new utilities extend existing routines to include error and data quality propagation. These are the utilities of greatest interest to the user community. Under this category are several tasks described in chapter 3 of the HST Introduction, **msarith**, **mscombine**, **msstatistics**, **msjoin** and **mssplit**, along with a few other tasks we describe below, **ndisplay**, **markdq**, **mosdisplay, pstack, pstats, sampinfo, nicpipe, biaseq, pedsky, pedsub, sampdiff and sampcum**. The first five are found in the package **toolbox.imgtools.mstools**; the remaining ones reside in the package **hst_calib.nicmos**.
2. ***Calibration-oriented utilities***. These tasks generate reference files, such as readnoise arrays, dark files, flatfields, non-linearity correction arrays, and bad pixel arrays, to feed the calibration database and to support the calibration pipelines. The tasks are designed specifically for the calibration of NICMOS and are not particularly useful for the general observer. The tools are **mstreakflat**, **msbadpix**, **ndark**, **nlincorr**, and **msreadnoise**. All are located in the calibration package **hst_calib.nicmos**.

The tasks in the **toolbox.imgtools.mstools** package are particularly useful for working with individual STIS and NICMOS imsets. See “Working with STIS and NICMOS Imsets” in chapter 3 of the HST Introduction if you are not familiar with these tasks. Below we describe a few tasks of specific interest to NICMOS observers. For additional details and examples of these and other tools, please refer to the online help.

Occasionally, the STScI NICMOS group introduces new software tasks outside the time frame of major new STSDAS releases. These are made available as a separate, add-on package called **nicproto**, and are usually announced in the *STScI Analysis News* (*STAN*) which is periodically distributed by e-mail (for more information, see section 1.1). This package can be obtained from the STSDAS or STScI NICMOS group WWW sites. Tasks in **nicproto** are generally somewhat experimental, and final versions migrate into the **hst_calib.nicmos** package in the next major STSDAS release. All routines described in this edition of the handbook are now available in the most recent STSDAS distribution, v2.3.

ndisplay and markdq

The **markdq** task reads the data quality (DQ) array from a NICMOS image and marks the DQ flags on the displayed image. Each flag value can be set independently to a different color or can be turned off. The **ndisplay** task combines the capabilities of the IRAF task **display** and the task **markdq**: it displays a NICMOS image and overlays the DQ flags according to a user-specified color-code. Both tasks are useful for locating specific DQ values, for example, the cosmic rays rejected by **calnica** in a MULTIACCUM image.

mosdisplay

The **mosdisplay** task provides a convenient way to display images from all IMSETS of a NICMOS MULTIACCUM image together as a mosaic in a single ximtool or saoimage window. The user may select which extension (e.g., `SCI, ERR, TIME, SAMP`, or `DQ`) to display, and can control the display threshold parameters or leave them to be automatically determined.

pstack and pstats

The **pstack** and **pstats** tasks plot all the samples of a specified pixel or image section, respectively, from a NICMOS MULTIACCUM image as a function of time. These tasks can be used to track the time behavior of an image on a pixel-by-pixel basis. For example, the temporal positions of cosmic ray hits or the onset of saturation during the course of an exposure can be located for a defined set of pixels. The pstats task can be particularly useful for identifying anomalous data behavior such as drifting bias levels or scattered light which may cause the background level to vary substantially during the course of an exposure.

sampinfo

The sampinfo task offers a convenient way to get readout-by-readout information about a NICMOS MULTIACCUM image. It provides information about the overall readout sequence (SAMP_SEQ, NEXTEND, NSAMP, and EXPTIME), and then for each imset of the multiextension fits file it lists the corresponding SAMPNUM, SAMPTIME and DELTATIME values. These can be useful bits of information when using non-standard processing techniques, such as the **biaseq** routine. An example of the use of **sampinfo** and its output is given in section 4.1.5.

nicpipe, biaseq, pedsky and pedsub

The **nicpipe** task provides a shortcut for partially processing NICMOS images through some but not all stages of **calnica**. Normally, this is meant for use in preparing images for "bias equalization" using the **biaseq** task (see chapter 4). Setting `stage=biaseq` takes the processing through the steps ZSIGCORR, ZOFFCORR, MASKCORR, BIASCORR, NOISCALC, DARKCORR, NLINCORR, and BARSCORR. Setting `stage=final` completes the processing. This can also be done by hand using **chcalpar** or **hedit** to change the processing control switches in the image headers (see section 3.5.2) and running **calnica** directly. Occasionally **nicpipe** can come in handy at other times besides **biaseq** processing: one example is given below.

The tasks **biaseq**, **pedsky**, and **pedsub** provide methods for dealing with the floating quadrant bias or "pedestal" effect in some (not all) data sets, and are described in section 4.1.5.

sampdiff and sampcum

The **sampdiff** task provides a convenient way to convert a MULTIACCUM image into a set of independent "first differences." Normally, each IMSET (readout) of a MULTIACCUM image is the cumulative sum of the total exposure time prior to that readout. As such, the `[sci,*]` images are not statistically independent. When analyzing NICMOS images, it is sometimes helpful to look at the data which was collected during each readout interval independent of that which was accumulated previously, i.e. by taking the difference of successive readouts. In this way, you can isolate readouts with problems (e.g., major cosmic ray hits or moving objects, sudden changes in bias, scattered light, etc.). The **sampdiff** task automates this process. *Note that, in general, this is only really a sensible thing to do if the image has not been converted from counts to count rate by the UNITCORR step of* ***calnica****!* The **sampcum** task inverts this process, re-accumulating the first differences.

Using These Tasks: An Example.

As an example, you might want to inspect NICMOS data for anomalies which occur during individual readouts during a MULTIACCUM using a procedure like this:

```
ni> nicpipe n4xj13jwq_raw.fits “” stage=biaseq

ni> sampdiff n4xj13jwq_ima.fits n4xj13jwq_fdiff.fits

ni> mosdisplay n4xj13jwq_fdiff.fits 1 extname=sci number+

ni> pstats n4xj13jwq_fdiff.fits[1:128,1:128]\
>>> extname=sci units=rate stat=midpt
```

In this example, the raw image is first partially processed through **calnica** using **nicpipe**. By setting `stage=biaseq`, the pipeline processing stops before flatfielding, conversion to count rate, and CRIDCALC cosmic ray processing. You then take first differences with **sampdiff**, displays them as a mosaic with **mosdisplay** to look for bars, bias jumps, monster cosmic rays, or other oddities in individual readouts, and then use **pstats** to plot the median count rate (`units=rate`) per sample time in the image quadrant `[1:128,1:128]`.

Having satisfied yourself that the data is fine, you might then complete processing using the **biaseq** and **pedsky** tasks in the following manner:

```
ni> biaseq n4xj13jwq_ima.fits n4xj13jwq_beq.fits \
>>> skysamps=1-13 fitbias+ fitjump+

ni> nicpipe n4xj13jwq_beq.fits “” stage=final

ni> pedsky n4xj13jwq_beq_cal.fits n4xj13jwq_ped.fits \
>>> salgorithm=auto interactive+ rmedian-
```

In this case, you have corrected readout-to-readout bias drifts with **biaseq**, also fitting for bias jumps in the process, and then carried out the final “pedestal” correction with **pedsky**, solving for the background level and pedestal interactively. Please see section 4.1.5 of this manual and the on-line help files for **biaseq** and **pedsky** for a detailed explanation of the parameters. The resulting end product image will be `n4xj13jwq_ped.fits`.

5.2 The NICMOS History Tool

When reducing and analyzing NICMOS data, it is sometimes important to know certain aspects about the instrument or telescope history at the time the observations were made. This information is not always available in the image headers or `*_spt.fits` files. The STScI NICMOS group has recently implemented a WWW-based NICMOS History Tool. This tool accepts the exposure start time or the dataset name as input, and provides options for retrieving information about the plate scale, focus, coronagraphic hole position, prior SAA passages, or the spacecraft attitude history. The spacecraft attitude history, for example, can be important for understanding the behavior of coronagraphic data (section 5.6), as is the coronagraphic hole position. The hole position is also helpful when reducing *non*-coronagraphic NIC2 data, as you may wish to mask out the hole when combining dithered data, or to know whether it has affected a target in your field (e.g., section 4.5.3). The SAA information can help you to pre-screen your data for frames which may be impacted by SAA-induced cosmic ray persistence (see section 4.6.2). The plate scale history is important for accurate astrometry and position measurements, since the scale of NICMOS varied somewhat with time as the dewar distortion changed (see section 5.4).

5.3 Photometric Calibrations

Atmospheric absorption bands limit the wavelength ranges over which one can usefully observe in the near-infrared from ground-based telescopes. NICMOS, however, can observe at any wavelength where its detectors are sensitive, and hence its filter bandpasses do not conform to those of instruments used at ground-based observatories. Instead, its filters were designed to meet the anticipated scientific demands. Thus in practice NICMOS does not have filters matched to any of the ground-based photometric bands. The photometric calibration of NICMOS data is discussed in this section. Cases of continuum sources, emission lines, and grism spectra will be presented.

5.3.1 Units for NICMOS Photometry

Given the multitude of units and systems that have been used for infrared photometry (magnitudes, Jy, W m^{-2} μm^{-1}, erg sec^{-1} cm^{-2} μm^{-1}, etc.) and given the lack of a standard for ground-based infrared filters, NICMOS has adopted the approach of calibration to physical units, i.e. Janskys (Jy), or Jy arcsec^{-2} for surface brightness. Details on how to transform different sets of units can be found in Appendix 2 of the *NICMOS Instrument Handbook* or obtained using the NICMOS Unit Conversion Tool (available on the STScI NICMOS web pages).

5.3.2 Fluxes and Magnitude Zeropoints

The NICMOS calibration pipeline provides two photometric parameters for the conversion of countrates into fluxes. These parameters are found in the keywords PHOTFNU and PHOTFLAM in the header of the calibrated image. The definitions of these keywords are discussed in section 3.3.4 of the HST Introduction. Very briefly, for NICMOS these are the bandpass-averaged flux densities (in F_λ for PHOTFLAM or F_ν for PHOTFNU) for a source that would produce a count rate of 1 DN sec^{-1}. PHOTFNU is given in units of Jy sec DN^{-1} and PHOTFLAM in units of ergs cm^{-2} Å^{-1} DN^{-1}. Because NICMOS calibrated data are given in countrate, i.e., DN sec^{-1}, the countrate to flux conversion is simply achieved by multiplying the countrate by the PHOTFNU or PHOTFLAM value, depending on which units are desired for the final calibrated image.

A fundamental challenge in calibrating NICMOS photometry is the inherent uncertainty in absolute flux calibration in the near-infrared, whether from the ground or from space. Unlike the situation at optical wavelengths, there really are no absolutely calibrated spectrophotometric flux standards at near-infrared wavelengths. Ground-based photometric calibration observations are mostly limited to relative magnitude determinations in more or less standard bandpasses (*JHK*) defined by atmospheric absorption windows. But the absolute flux calibration of primary infrared standards has generally been based on reference to some assumption that a particular type of standard star, e.g., solar-type stars, A0 stars, or white dwarfs, has a spectrum similar to something else that is believed to be well understood, such as the Sun or a well-calibrated stellar atmosphere model. As an example, see Campins, Rieke & Lebofsky (1985), who used the solar analog approach to derive absolute flux calibrations in the near-infrared. These calibrations are uncertain at a level of several percent, setting a fundamental limit on the accuracy of any absolute flux calibration in the near-infrared, including that of NICMOS.

In the case of NICMOS, the photometric calibration is based on observations of solar analogs and hydrogen white dwarfs (WDs). The stars P330E (solar analog) and G191B2B (WD) define the primary calibration, with a few other stars observed as cross-checks. Bohlin, Dickinson & Calzetti (2001) have recently recalibrated the absolute flux density distribution of the HST NICMOS standards. The optical and UV portions of their spectra were measured with high quality HST STIS observations, which were then carefully matched to stellar atmosphere models for extrapolation to near-infrared wavelengths. The infrared extrapolations were then checked by comparison to the most up-to-date ground-based broad band photometry for these stars, normalized to flux density using the Campins et al. (1985) calibrations (adjusted for small bandpass differences).

The STScI NICMOS group and NICMOS IDT have recently reanalyzed all photometric calibration data taken in orbit with NICMOS during Cycles 7 and 7N. Unlike earlier efforts, the new work uses a larger number of measurements, applies uniform and up-to-date NICMOS calibration software and reference files, and employs a more careful treatment of the NICMOS photometric aperture corrections[1] than was used for the original calibration effort. Table 5.1 through table 5.3 give the current best estimates of the PHOTFNU and PHOTFLAM values for NICMOS filters. These new values update previous numbers given in older versions of the PHOTTAB calibration reference files, as well as an interim set of calibrations presented in an earlier edition of this *Handbook* (version 4.0). The new calibrations are believed to be accurate to better than <5%, thus satisfying the absolute calibration requirements for NICMOS. For most filters, the *relative* calibration accuracy should be better, <3%, with the possible exception of some narrow band filters where uncertainties in

[1] The PHOTFNU and PHOTFLAM keywords represent photometry corrected to infinite aperture, i.e., the scaling between stellar flux density and the *total* count rate measured from a star within a (hypothetical) infinite aperture. The actual standard star measurements are made within a fixed, finite radius aperture and corrected to infinite aperture using mode point spread functions from **TinyTim**.

absorption line strengths for the solar analog and white dwarf models may result in greater calibration errors.

Please note that the NIC1 and NIC2 polarizers have not yet been recalibrated. The values given in table 5.1 through table 5.3 are taken from the older PHOTTAB calibration reference file `i7l12297n_pht.fits`, and are marked by asterisks. Note also that the phtometric calibration for the extremely broad band filters F140W, F150W and F175W should be used with caution. These bandpasses are so broad that the conversion from source flux to count rate will be very strongly dependent on the color or spectrum of the source.

The photometric calibration information presented here is appropriate for NICMOS data taken during Cycles 7 and 7N only. In Cycle 11 and thereafter, NICMOS will most likely be operating at a significantly warmer temperature, regulated by the NICMOS Cooling System. Because the detector quantum efficiency is a function of temperature, the photometric zeropoint calibrations will almost certainly be different than those appropriate to Cycle 7 data. New photometric calibrations will be derived as quickly as possible in Cycle 11. Please check the STScI NICMOS web pages for updates on photometric calibration for Cycle 11 data.

In the header of calibrated NICMOS images, there are three additional photometric parameters that characterize the filter used for the observation (`PHOTPLAM` and `PHOTBW`) and provide the ST magnitude zero point (`PHOTZPT`). `PHOTPLAM` gives the value of the pivot wavelength of the filter in Angstroms. This wavelength is source-independent and is the wavelength for which:

$$PHOTFLAM = c \times PHOTFNU \times PHOTPLAM^{-2}$$

where c is the speed of light in vacuum. `PHOTBW` gives the rms band width of the filter in Å (see the *Synphot User's Guide* for a detailed definition of both parameters).

The magnitude of an object can be determined in the ST system (e.g., based on a constant flux per unit wavelength) using the photometric zero-point keyword `PHOTZPT` (= -21.1) simply by:

$$m_{ST} = -2.5\log(PHOTFLAM \times CR) + PHOTZPT$$

$$= -2.5\log(PHOTFLAM \times CR) - 21.1$$

where *CR* is the count rate in units of DN sec^{-1}. On the other hand, the magnitude in Oke's AB$_\nu$ system (e.g., based on a constant flux per unit frequency) is obtained by applying the following expression:

$$m_{AB} = -2.5\log(10^{-23} \times PHOTFNU \times CR) - 48.6$$

$$= -2.5\log(PHOTFNU \times CR) + 8.9$$

As noted above, the NICMOS filter bandpasses do not match those of ground-based filters, and therefore it is not straightforward to derive "standard" infrared magnitudes normalized, e.g., to the spectral energy distribution of Vega. However, it may sometimes be useful to approximate a Vega-based photometric system by using an estimate of the flux density of Vega through the NICMOS bandpasses as a photometric zeropoint. Because there are no direct spectrophotometric observations of Vega or other A0V standards through the NICMOS filter system, our best option is to use a spectrophotometric model for Vega and synthetically integrate it through the bandpasses. Table 5.1 through table 5.3 give bandpass averaged flux densities (in Jy) for the NICMOS filters using a model reference spectrum of Vega (Colina, Bohlin & Castelli 1996, ISR CAL/SCS-008). An approximate Vega-normalized magnitude may be computed as

$$m = ZP(\text{Vega}) - 2.5 \log (PHOTFNU \times CR \times \langle F_\nu(\text{Vega})\rangle^{-1})$$

where *ZP*(Vega) is the magnitude of Vega. Under the "CIT" infrared photometry scale, Vega is defined to have ZP(Vega) = 0.0, whereas in the "Arizona" system (e.g., Campins et al. 1985, *AJ*, 90, 896), Vega has *ZP*(Vega) = 0.02.

Table 5.1: NIC1 Photometric Zeropoints

Spectral Element	PHOTFLAM (erg cm^{-2} A^{-1} DN^{-1})	PHOTFNU (Jy sec DN^{-1})	$\langle F_\nu(\mathrm{Vega})\rangle$ (Jy)
F090M	4.883E-18	1.332E-05	2156.1
F095N	7.397E-17	2.244E-04	1733.7
F097N	5.847E-17	1.841E-04	2217.8
F108N	3.994E-17	1.559E-04	1881.0
F110M	1.743E-18	7.068E-06	1820.7
F110W	5.545E-19	2.358E-06	1773.7
F113N	2.967E-17	1.263E-04	1768.0
F140W	2.027E-19	1.402E-06	1304.7
F145M	8.066E-19	5.702E-06	1195.8
F160W	2.901E-19	2.499E-06	1039.3
F164N	6.161E-18	5.567E-05	965.4
F165M	5.497E-19	4.985E-06	985.1
F166N	6.087E-18	5.600E-05	1010.0
F170M	4.766E-19	4.631E-06	945.4
F187N	4.265E-18	5.000E-05	772.9
F190N	4.127E-18	4.962E-05	805.5
POL0S	1.866E-18*	6.996E-06*	1896.1
POL120S	1.843E-18*	6.912E-06*	1896.1
POLS240S	1.844E-18*	6.914E-06*	1896.1

*These values have not been recalibrated, and are taken from PHOTTAB `i7l12297n_pht.fits`

Table 5.2: NIC2 Photometric Zeropoints

Spectral Element	PHOTFLAM (erg cm^{-2} A^{-1} DN^{-1})	PHOTFNU (Jy sec DN^{-1})	$\langle F_\nu$(Vega)$\rangle$ (Jy)
F110W	4.382E-19	1.861E-06	1775.0
F160W	2.401E-19	2.066E-06	1040.7
F165M	4.541E-19	4.132E-06	982.8
F171M	1.061E-18	1.048E-05	930.7
F180M	1.012E-18	1.091E-05	870.9
F187N	3.578E-18	4.191-05	774.3
F187W	3.193E-19	3.732E-06	816.5
F190N	3.500E-18	4.215E-05	804.7
F204M	5.526E-19	7.637E-06	716.2
F205W	8.599E-20	1.231E-06	703.6
F207M	3.793E-19	5.486E-06	686.8
F212N	2427E-18	3.643E-05	664.7
F215N	2.553E-18	3.932E-05	645.1
F216N	2.327E-18	3.636E-05	605.9
F222M	3.177E-19	5.214E-06	610.4
F237M	2.406E-19	4.507E-06	546.1
POL0L	5.711E-19*	7.626E-06*	734.1
POL120L	5.639E-19*	7.530E-06*	734.1
POL240L	5.630E-19*	7.517E-06*	734.1

*These values have not been recalibrated, and are taken from PHOTTAB `i7l12297n_pht.fits`

Table 5.3: NIC3 Photometric Zeropoints

Spectral Element	PHOTFLAM (erg cm^{-2} A^{-1} DN^{-1})	PHOTFNU (Jy sec DN^{-1})	$\langle F_\nu(\text{Vega})\rangle$ (Jy)
F108N	4.283E-17	1.667E-04	1889.8
F110W	6.143E-19	2.600E-06	1780.2
F113N	3.220E-17	1.368E-04	1771.7
F150W	1.766E-19	1.412E-06	1157.7
F160W	3.125E-19	2.694E-06	1038.9
F164N	6.663E-18	6.021E-05	965.4
F166N	6.793E-18	6.232E-05	1011.2
F175W	8.529E-20	9.590E-07	908.3
F187N	4.679E-18	5.486E-05	772.9
F190N	4.306E-18	5.187E-05	804.7
F196N	3.735E-18	4.805E-05	757.3
F200N	3.493E-18	4.649E-05	739.7
F212N	3.008E-18	4.515E-05	664.7
F215N	3.179E-18	4.896E-05	645.1
F222M	3.927E-19	6.446E-06	610.4
F240M	2.161E-19	4.140E-06	534.5

5.3.3 Photometric Corrections

Photometric Stability with Time and Temperature

The photometric stability of NICMOS was monitored throughout the instrument's lifetime in Cycles 7 and 7N using repeated observations of the primary standard stars through a limited set of filters in each camera. There is some evidence for a small drift in the throughout with time, amounting to roughly 2% over the lifetime of the instrument. This is to be expected, as the array quantum efficiency is temperature dependent, and the instrument temperature gradually warmed throughout Cycle 7 and 7N. At present, this photometric "drift" is no larger than other known uncertainties in the absolute photometric calibration of NICMOS, and we do not discuss it further here. In the future, STScI may provide information for correcting the photometric zeropoints as a function of the observation date. This will be announced via the NICMOS WWW pages and the *Space Telescope Analysis Newsletter (STAN)*. It is important to note, however, that NICMOS photometric zeropoints will almost certainly be significantly different in Cycle 11 and beyond when the instrument is operating at a warmer temperature with the NCS.

Differential Photometry

The photometric values provided in the headers are obtained from measurements of standard stars in the central regions of the detectors. Both high frequency (pixel-to-pixel) and low frequency (large-scale structures) sensitivity variations are corrected using on-orbit flats. On-orbit differential photometry characterization of NICMOS cameras 1 and 2 indicate that residual, large scale deviations amount to < 2% (rms). Much larger variations are seen for Camera 3, but are due to a different effect, intrapixel sensitivity variations, which we will discuss next.

Intrapixel Sensitivity Variations and Camera 3

As with many other infrared arrays, NICMOS detector sensitivity varies across the area of each physical pixel. It is higher in the center, and lower near the edges. In practical terms, this means that for a source whose flux changes rapidly on a scale comparable with or smaller than that of a pixel, the measured countrate, and therefore the derived flux, will depend on where the center of the source lies with respect to the center of the pixel.

For NICMOS cameras 1 and 2, the PSF is sufficiently well sampled that intrapixel sensitivity variations do not introduce a large (or at least, not a dominant) effect on point source photometry. Camera 3 images, however, are highly undersampled, and the exact position of a point source relative to the pixel grid can introduce large variations in the measured signal. Measurements made using calibration test data, as well as science data with large numbers of dither positions, show that the flux variation can be as large as 30%. The tighter the PSF is, the larger these variations will be. For that reason, “in focus” images taken during the two Cycle 7N refocus campaigns are more subject to the impact of intrapixel sensitivity variations, although the effect is still appreciable for non-campaign data as well. Similarly, stellar images through the shorter wavelength filters (where the telescope diffraction limit is smaller) show a greater variation due to this effect than do longer wavelength data. In addition, the intrapixel sensitivity variations can introduce complications for measuring image centroids, “pulling” the centroid away from pixel corners and edges toward the center of the pixel.

The effects of intrapixel sensitivity variations and possible approaches to correcting NIC3 point source photometry are discussed in two useful references: NICMOS ISR-99-005 (Storrs et al.), available from the STScI WWW pages, and a paper by Tod Lauer (1999, PASP 107, p.1434). Storrs et al. derive a photometric correction for Camera 3 F110W and F160W stellar images based on a “sharpness ratio,” defined as the ratio of the flux in the peak pixel to that in the integrated PSF. This method is simple to apply and avoids uncertainties due to centroiding errors. Lauer derives a two dimensional pixel response map which can be used to correct photometry if the position of a point source can be measured accurately.

Intrapixel sensitivity variations should have substantially less effect on spatially extended sources. In general, given well dithered data sampling many different sub-pixel positions, the variations due to intrapixel sensitivity should average out, but given the rather large amplitude of the effect for point sources (especially for short wavelength, in-focus NIC3 observations), many dither positions (>10) are probably needed before the error on the mean count rate is smaller than other sources of photometric uncertainty in NICMOS.

Camera 3 intrapixel sensitivity variations can have a major impact on photometry of point sources. For data taken with fewer than 10 dither positions, this uncertainty may dominate photometric errors unless steps are taken to correct it. We strongly recommend that Camera 3 users concerned with photometry of point sources read the references cited here.

Temporal and Spatial PSF and Focus Variations

The point spread function (PSF) of HST changes with time, and these changes will affect photometry using small apertures. Changes in focus observed on an orbital timescale are due mainly to thermal breathing of the telescope. In addition, there were long-term variations in NICMOS focus as the cryogen evaporated and the dewar relaxed. The solid nitrogen introduced stress on the instrument, and NICMOS detectors moved along the focus direction as this stress diminished. The change was especially rapid during the SMOV period. The focus was largely stable thereafter, however, throughout most of the instrument lifetime.

The dewar distortion had the greatest effect on the focus of Camera 3, pushing it beyond the range of the Pupil Alignment Mechanism (PAM). For this reason, two special Camera 3 refocus campaigns, each three weeks long, were undertaken, one from 12 January through 1 February 1998, and the second from 4 June through 28 June 1998. During these periods, the HST secondary mirror was moved in order to bring Camera 3 into the range where its correct focus could be reached by the PAM.

The focus history of the telescope at the time of your observation can be determined using the NICMOS History Tool (see section 5.2).

In addition, the NICMOS PSF varies somewhat as a function of position on the array, and this can affect photometry made either with small apertures (by changing the effective aperture correction as a function of position), or using PSF fitting programs such as **daophot** unless there are enough stars in the field for the program to derive a positionally dependent

PSF. The focus variations, however, are relatively small (see NICMOS ISR 98-005, "NICMOS Focus Field Variations and Focus Centering," Suchkov & Galas 1998). We have used synthetic PSFs from **TinyTim** (see "Aperture Correction" on page 5-15) to determine the positional dependence of the aperture correction as a function of position on NIC2, where the focus variations are largest. For F110W (the effect should be largest at short wavelengths), the maximum to minimum variation in the aperture correction for photometry measured within a 2 pixel radius is < 2%. The variations are expected to be smaller for the other cameras and for images at longer wavelengths, although this has not yet been empirically verified.

Aperture Correction

It is often difficult to measure the total flux of a point source due to the extended wings of the PSF, diffraction spikes, and scattered light. Such measurements are particularly difficult in crowded fields where the extended wings of sources can overlap with each other. An accurate method of measuring the integrated flux in these situations could consist of several steps:

1. Measure in the image the total counts within a small radius.
2. Simulate the **TinyTim**[2] PSF for the particular camera-filter combination and position in the detector. **TinyTim** allows the user to simulate PSFs with various focus settings, and users concerned with small focus variations should be careful to set these parameters to the values appropriate for their observations. (This will have the largest effect for NIC3 data taken outside the refocus campaigns.)
3. Use the simulated PSF to measure the fraction of total flux within the photometry aperture.

To obtain the total flux of the source, the countrate then only needs to be multiplied by the `PHOTFNU` or `PHOTFLAM` value and by the inverse of the measured fraction obtained in step three above.

Empirical PSFs could also be used for the above mentioned method. There was no special Cycle 7 calibration program to obtain PSF measurements for all camera and filter combinations. A significant amount of PSF data does exist in the archive, however: empirical PSFs for the central regions of the detectors can be obtained from the calibrated images obtained for the Cycle 7 absolute photometry (proposal 7691) and photometric monitoring (proposal 7607) programs. The STScI NICMOS group will make a library of empirical PSFs available at some time in the future.

[2] **TinyTim** software can be retrieved from the Web at: http://www.stsci.edu/software/tinytim/tinytim.html

Out-of-Band Leaks

Many very red targets (e.g., protostars) were observed with NICMOS at short wavelengths (~1 µm). For these sources the flux at ~2.2–2.5 µm could be orders of magnitude larger than at ~1.0 µm and therefore exceptionally good out-of-band blocking would be required. Pre-launch tests indicated that for very red sources (temperature ~700 K and lower), some NICMOS filters *might* have significant red leaks. The suspect filters were F090M, F095N, F097N, F108N, F110M, F110W, F113N, F187N and F190N. A limited set of in-flight tests (in calibration programs 7691 and 7904) were made during Cycle 7, observing very red stars to calibrate the possible effects of out-of-band leaks. These data have never been carefully analyzed, but preliminary indications are that actual red leaks are insignificant or non-existent. We nevertheless caution here that this analysis is only preliminary. Users interested in photometry of sources with extreme colors should check the STScI NICMOS WWW pages for updates, or examine the in-flight test photometry of red stars themselves.

Non-Zero Zeroth Read Correction for Bright Sources

The problem of the non-zero zeroth read for bright sources was discussed in section 3.3 (ZSIGCORR) and section 4.3.2. If significant signal from an object is present in the zeroth read, then this was not properly taken into account in the nonlinearity corrections made for data processed by the OPUS pipeline before 11 November 1997. It is advisable to reprocess the data with the most recent version of the calibration software which includes an additional step (`ZSIGCORR`) to account for this zero read signal correction. When reprocessing such data, **calnica** version 3.3 or greater will automatically apply this step to all MULTIACCUM mode observations if both `ZOFFCORR` and `NLINCORR` are also being performed. All data retrieved from the Archive via OTFR after 26 September 2001 are automatically processed with this step.

Magnitudes and Photometric Systems Transformations

As was previously mentioned, NICMOS data are calibrated in units of Jy or Jy arcsec^{-2} for flux densities and surface brightnesses, respectively. The filter bandpasses do not exactly match those of ground-based JHK filters, and therefore if you wish to derive standard JHK magnitudes from NICMOS data, then it will be necessary to apply a color term correction. The recommended NICMOS JHK-analog system is obtained using the F110W, F160W and F222M filters, although the F110W filter especially is quite different from the standard J bandpass (specifically, it is much broader and bluer). Synthetic color terms can be computed using **synphot**, given real or model spectra for an astronomical source and the NICMOS and ground-based filter bandpasses (see, for example, Holtzman et al. 1995, PASP 107, 1065, where a similar procedure was used to derive color terms for WFPC2 photometry).

Also, as part of the Cycle 7 absolute photometry program, we observed a few blue stars (white dwarfs), intermediate color stars (solar analogs), and very red stars covering a large range in color (table 5.4). The calibrated data are available from the archive for users who wish to derive their own color terms to transform their HST fluxes into any ground-based system. The ground-based photometry given in the table below is based on preliminary and incomplete measurements, and will be updated when the final NICMOS photometric calibrations are completed.

Table 5.4: List of Stars for Photometric Transformations

Name	H	J-H	H-K	Status	Program IDs
G191-B2B	12.6	-0.10	-0.14	Primary standard (white dwarf).	7691, 7816
P330E	11.6	0.28	0.07	Primary standard (solar analog).	7691, 7816
OPH-S1	7.3	1.53	0.94	Primary standard (red standard).	7049, 7691
GD153				Secondary standard (white dwarf).	7904, 7816
P177D	12.0	0.28	0.06	Secondary standard (solar analog).	7904, 7816
CSKD-12	9.5	2.08	0.89	Secondary standard (red standard).	7904, 7816
BRI0021	11.1	0.75	0.52	Secondary standard (red standard).	7904, 7816

5.3.4 Absolute Photometry for Emission Line Filters

The narrow band filters in NICMOS are intended primarily for observations of emission or absorption lines in sources. Because the photometric conversion factors PHOTFNU and PHOTFLAM for all NICMOS filters are obtained from continuum observations of emission-line free standard stars, the flux in erg sec^{-1} cm^{-2} of an emission line is given by the expression:

$$Flux_{line} = 1.054 \times FWHM \times PHOTFLAM \times CR$$

where FWHM is the full width half maximum of the equivalent gaussian filter to the narrow-band filter used (see Appendix 1 of the *NICMOS Instrument Handbook*), and we have assumed that the continuum has been already subtracted from the total flux in the filter and that the line is centered in the filter. If the emission line is not at the central wavelength of the filter, the line flux will need correction for the filter transmission curve. To estimate the variation in the absolute flux due to the positioning and width of the emission line in the filter bandpass, the **synphot** task **calcphot** can be used as shown below. See the *Synphot User's Guide* for additional help.

Figure 5.1: Estimating Absolute Flux Variation

```
sy> epar calcphot
obsmode = nicmos,3,f212n,dn Instrument observation mode
spectrum= gauss(21200,40)*unit(1E-13,flam) Synthetic spectrum to calculate
form = counts Form for output data
(func = effstim) Function of output data
(vzero = ) List of values for variable zero
(output = none) Output table name
(append = no) Append to existing table?
(wavetab= ) Wavelength table name
(result = 0.) Result of synphot calculation for form
(refdata= ) Reference data
(mode = a)

sy> epar calcphot
obsmode = nicmos,3,f212n,dn Instrument observation mode
spectrum= gauss(21280,40)*unit(1E-13,flam) Synthetic spectrum to calculate
form = counts Form for output data
(func = effstim) Function of output data
(vzero = ) List of values for variable zero
(output = none) Output table name
(append = no) Append to existing table?
(wavetab= ) Wavelength table name
(result = 0.) Result of synphot calculation for form
(refdata= ) Reference data
(mode = a)
```

The examples in figure 5.1 compute the countrate in the NIC3 F212N filter for a H_2 (2.12 micron) emission line having a gaussian profile of 40 Angstroms and a peak flux of 1.0 x 10^{-13} erg sec^{-1} cm^{-2} A^{-1}. The integrated flux will then be 4.2 x 10^{-12} erg sec^{-1} cm^{-2}. In the first example the H_2 emission line is at zero redshift and centered on the filter while in the second example the line is redshifted by 80 Angstroms. If the emission line is centered on the filter, the H_2 flux will produce 7421.1 DN sec^{-1} while the countrate will be ~90% of this value (i.e., 6662.3 DN sec^{-1}) for the redshifted emission line. The expression for the $\mathrm{Flux}_{\mathrm{line}}$ above can be directly applied to the first case, while a correction factor 1.11 = (7421.1/6662.2) is needed in the second case.

5.3.5 Absolute Spectrophotometry with NICMOS Grisms

The accuracy of the absolute spectrophotometry with NICMOS grisms depends on three limiting factors:

- Accuracy of the spectral energy distribution of the standard stars used to obtain the inverse sensitivity curve.
- Quality of the flat-fielding and background subtraction of the calibration observations.
- Quality of the flat-fielding and background subtraction of the science observation itself.

The major source of uncertainty in the absolute spectrophotometry of a given source comes from the variability and structure of the background. Every pixel on the NIC3 array will receive background radiation over the entire spectral bandpass of the particular grism, while the source spectrum will be dispersed. The accuracy of the background subtraction is limited by our knowledge of the spectral response of each pixel, which is somewhat different from pixel to pixel. Extracted spectra will have to be corrected for the spectral response of each pixel. The accuracy of this correction is again limited by our knowledge of the detector response.

Remember that the spectra are not aligned with the X axis of the detector, and so move from one row to the next. This makes grism spectra especially sensitive to the intrapixel sensitivity variation. The **NICMOSlook** software (see section 5.8) version 2.7.4 and higher has a capability built in to correct for this variation. It depends critically on the placement of the spectrum, however, and the software handbook describes an iterative procedure to determine the best correction for this problem.

The absolute flux calibration of the spectral energy distribution of the standard stars in the 0.8 to 2.5 µm wavelength range is known to 2–5%. Characterization of the grisms' absolute sensitivity during SMOV indicates that the absolute calibration of grism spectrophotometry for bright sources, i.e. well above background, will have a total 20–30% uncertainty. For Grism C the uncertainties could be even higher because of the large thermal background in this wavelength range (1.4 to 2.5 µm).

Grism data reduction and calibration are discussed in more detail in section 5.8.

5.4 Astrometry, Pixel scales, and Geometric Distortion

Accurately transforming pixel positions in NICMOS images to relative or absolute astronomical coordinates involves several considerations. Here we consider each of these in turn.

5.4.1 Pixel Scale Time Dependence

The distortion of the NICMOS dewar due to the cryogen expansion affected the optical path, pushing the cold well and cameras closer to the field divider assembly and cold mask. This caused the effective focal length, and hence the pixel scale, to change somewhat as a function of time during the lifetime of the instrument. The changes were most rapid during the orbital verification period and approximately the first 100 days of instrument operations, then slowed, with only small variation thereafter. The pixel scale was monitored regularly throughout the lifetime of the instrument. A tabular and graphical record of these measurements for all three cameras can be found on the STScI NICMOS WWW pages at:

`http://hst.stsci.edu/nicmos/performance/platescale`.

The NICMOS history tool (see section 5.2) will also provide the scale information for any given observation. These sources should be consulted before transforming pixel coordinates in the image to arcseconds.

5.4.2 X and Y Pixel Scale Differences

The NICMOS arrays are slightly tilted relative to the focal plane, and therefore the NICMOS pixel scales along the X and Y axes of each camera are slightly different, i.e., projected on the sky, the pixels are actually slightly rectangular, not square. This is a small effect: the X/Y scale ratios S_X/S_Y are approximately 1.0039, 1.0086, and 1.0030 for cameras 1, 2 and 3, respectively, in the sense that the X scale in arcseconds per pixel is larger than the Y scale in each case.[3] It amounts to 1 to 2 pixels "extra width" in the X direction relative to Y over the field of view of the cameras, and for precision astrometry or when registering NICMOS images to data taken with other instruments such as WFPC2, it may be important to take the effect into account. This can easily be done when drizzling NICMOS images using the coefficient matrices presented below.

5.4.3 Geometric Distortion

The distortion corrections for the NICMOS cameras are small, but for precision astrometry or when registering and coadding images (especially taken with a wide dither pattern, or when assembling mosaics covering a large field) it should be taken into account. The geometric distortion was measured using dithered observations of the astrometric field NGC1850 through all three cameras. The data analysis and results are described in detail in an instrument science report (ISR) by Cox et al. 1997 (ISR OSG-CAL-97-07). As noted in that report, the original observations were made in May 1997 when NICMOS Camera 3 was not in perfect focus (which can only be achieved during special campaigns when the telescope secondary is moved). This, together with field vignetting, means that the Camera 3 distortion solution from the Cox et al. ISR is not as well determined as that for Cameras 1 and 2. The geometric distortion correction for Camera 3 was rederived later using a reobservation made during the January 1998 refocus campaign; the Field Offset Mirror (FOM)

[3] Please note that these scale ratios may not agree precisely with the values reported in the NICMOS pixel scale records described above. This is because they are defined and derived in a slightly different way. The values given here are the scale ratios derived from the geometric distortion solution described in section 5.4.3, and formally apply to the aspect ratio of the pixels at array center, position (128,128). These are the appropriate values to use in combination with the geometric distortion corrections. The X and Y scales reported by the NICMOS History Tool are, in effect, mean values over the array, averaging over the scale changes induced by geometric distortion.

was also moved to substantially reduce the vignetting. Cox et al. have derived an improved Camera 3 distortion solution which has not been presented in any other NICMOS document, but which we describe here. For Camera 3, these values therefore supersede those from the original ISR by Cox et al.

The maximum total positional deviations (in pixels), assuming zero distortion at field center, are approximately 0.9, 0.25, and 0.75 pixels for Cameras 1, 2 and 3, respectively. Cox et al. found that quadratic relations were adequate for describing the distortion given the accuracy of the observational measurements. Given a pixel position (x,y), we define new coordinates (x',y') relative to the array center, here chosen to be at pixel 128,128:

$$x' = x - 128,$$
$$y' = y - 128.$$

The positions corrected for distortion, (x_c',y_c'), are then given by

$$x_c' = a_{10}\, x' + a_{11}\, y' + a_{20}\, x'^2 + a_{21}\, x'y' + a_{22}\, y'^2$$
$$y_c' = b_{10}\, x' + b_{11}\, y' + b_{20}\, x'^2 + b_{21}\, x'y' + b_{22}\, y'^2$$

where the origin of the (x_c', y_c') coordinate system is chosen to be coincident with that of (x', y'), i.e. pixel (128,128) of the array. As defined by Cox et al., the distortion corrections explicitly do not account for the X and Y scale difference described above, and therefore a_{10} and b_{11} are fixed at 1. Also, a_{11} (the linear dependence of x on y) is held at 0, although b_{10} is derived in the fit. This fixes the y axis orientation, but allows for departures from orthogonality. The fitted coefficients for the three cameras are given in table 5.5.

Table 5.5: NICMOS Geometric Distortion Coefficients

	NIC 1		**NIC 2**		**NIC3**	
Coefficient	value	±	value	±	value	±
a_{10}	1.0	0.0	1.0	0.0	1.0	0.0
a_{20}	10.9e-6	1.4e-6	-9.98e-6	0.85e-6	2.78e-6	2.59e-6
a_{21}	-10.6e-6	5.9e-6	-1.17e-6	0.26e-6	17.3e-6	6.02e-6
a_{22}	7.4e-6	2.6e-6	-4.45e-6	0.71e-6	6.02e-6	2.05e-6
b_{10}	48.5e-4	1.7e-4	-2.97e-4	0.53e-4	-16.3e-4	0.12e-4
b_{11}	1.0	0.0	1.0	0.0	1.0	0.0
b_{20}	5.0e-6	1.4e-6	1.02e-6	0.82e-6	-15.44e-6	1.75e-6
b_{21}	10.7e-6	4.5e-6	-2.57e-6	0.73e-6	-3.34e-6	3.84e-6
b_{22}	14.4e-6	2.1e-6	-0.15e-6	1.27e-6	4.26e-6	2.13e-6

After applying this distortion correction, remember to add 128 to each of the x_c' and y_c' values to transform the origin back to pixel position (128,128).

Also remember that the above solution does *not* correct the X/Y scale differences described above. The preferred way to do this would be to multiply all the *a* coefficients in the expression above by the square root of the pixel scale ratio, $\sqrt{}(S_X/S_Y)$, and the *b* coefficients by the reciprocal value, $\sqrt{}(S_Y/S_X)$, using the scale ratios given above. This will correct all pixels to square geometry, preserving the area of a pixel located at array center. This is the approach used in the distortion coefficients used by the **drizzle** algorithm (see section 5.4.4).

Cox et al. also solve for the detector y axis orientation relative to the telescope V3 axis, finding it to be -44.542± ± 0.006, -45.459 ± 0.002, and -45.037 ± 0.005 degrees for Cameras 1, 2 and 3, respectively.

5.4.4 Drizzling

One way to apply the geometric distortion correction when combining dithered NICMOS images is to use the **drizzle** software, which is incorporated into the package **stsdas.analysis.dither.** This software was written to allow geometric distortion corrections to be applied during drizzling. In order to do so, it is necessary to specify a coefficient file with the parameter `drizzle.coeffs`. Geometric distortion coefficient files in a format suitable for use with **drizzle** are available from the STScI NICMOS web pages and are also provided with the **stsdas.analysis.dither** package. In these, the coefficients from Cox et al. have been multiplied by the square root of the pixel scale ratio so that the output images will have square pixels with uniform X and Y scales, as described above. The absolute pixel scale (arcseconds per pixel) is not set by the drizzling procedure, and should be determined by consulting the NICMOS History Tool, as explained above. The appropriate value to use with the drizzled images is the geometric mean of the X and Y scales specified by the web pages or History Tool, i.e. $\sqrt{}(S_X \times S_Y)$.

5.4.5 Absolute Astrometry

As with all HST instruments, determining the absolute astrometry in NICMOS images to good accuracy requires precise astrometry from some external source for some object or objects within the field of view. The absolute pointing reference given by world coordinate system (WCS) information encoded in NICMOS image headers is derived from the HST Guide Star Catalog (GSC) positions for the stars used to guide the observations. Although the GSC positions, and hence the WCS astrometry, are generally good, deviations by as much as 2 arcseconds from an absolute celestial reference frame (e.g., FK5) are not uncommon. The WCS information provided with NICMOS images does account (at least to first order) for the nominal pixel scale, camera orientation, and even the X and Y pixel scale differences of the cameras. It does not, however, include any higher order corrections for geometric distortion. Therefore relative positions measured using the WCS (e.g., with the **IRAF** tasks **imexam** or **rimcur**) are fairly accurate, even if the absolute pointing reference may not be exact. For precision relative astrometry, you should recheck the exact pixel scale using the information from the NICMOS History Tool, and take the geometric distortion of the Cameras into account.

Finally, it is important to note that precise astrometry with NIC3 can be complicated by three effects. First, for data taken outside the January or June 1998 refocus campaigns, focus effects may affect the geometry of the focal plane. Second, data taken prior to January 1998 was usually vignetted (see section 4.5.4), which can also distort the geometry. Finally, intrapixel sensitivity variations (see section 5.3, and NICMOS ISR-99-005, Storrs et al.) can affect centroiding for undersampled point source images.

5.5 PSF Subtraction

Accurate PSF subtraction is a prime concern for an observer wishing to study faint features around bright objects. Typical situations are: a host galaxy harboring a bright quasar; circumstellar nebulosity around a bright star; faint companions of a bright star, etc. PSF subtraction for NICMOS data can be very effective, especially for images from cameras 1 and 2, thanks to a few important features of this instrument:

- Cameras 1 and 2 provide images which are diffraction limited, and well sampled (see figure 5.2 and figure 5.3).
- The MULTIACCUM mode automatically provides sub-exposures with different exposure times, and **calnica** processing will use only unsaturated data values when constructing the calibrated image.
- For a saturated bright central point source, there is no blooming along the detector array columns as there is with a CCD.

One way to get a high quality PSF for subtraction is to measure an isolated bright unsaturated star in the same image or to construct a composite PSF using good stars in the image. This can be accomplished using the IRAF **digiphot** package as described in *A User's Guide to Stellar CCD photometry with IRAF*[4]. A PSF obtained from the same image

[4] This and other relevant IRAF documents can be obtained from the IRAF Web site at:http://iraf.noao.edu/.

ensures that effects of telescope focus and pointing jitter on the image quality are properly taken into account. In particular, this approach will take care of the *breathing* effect: variations of focus position due to thermally induced mechanical displacements in the HST optical path. One disadvantage of this method, however, is that it is difficult to account for positional variations in the PSF over the field of view of the NICMOS cameras (see discussion in section 5.3.3, and also in Suchkov & Galas 1998, NICMOS ISR 98-005). Although these variations do not have a large affect on NICMOS aperture photometry, they can be large enough to cause problems for PSF matching and subtraction. If no suitable star can be found in the image, one can resort to a synthetic PSF computed using **TinyTim**.

Figure 5.2: NIC1 Image with PSF Radial Profile

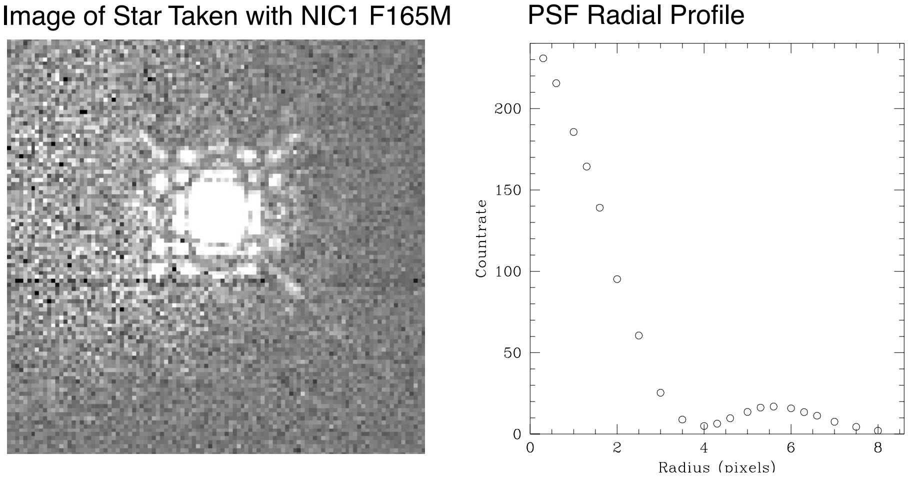


Image of Star Taken with NIC1 F165M

PSF Radial Profile

Figure 5.3: NIC2 Image with Radial Profile

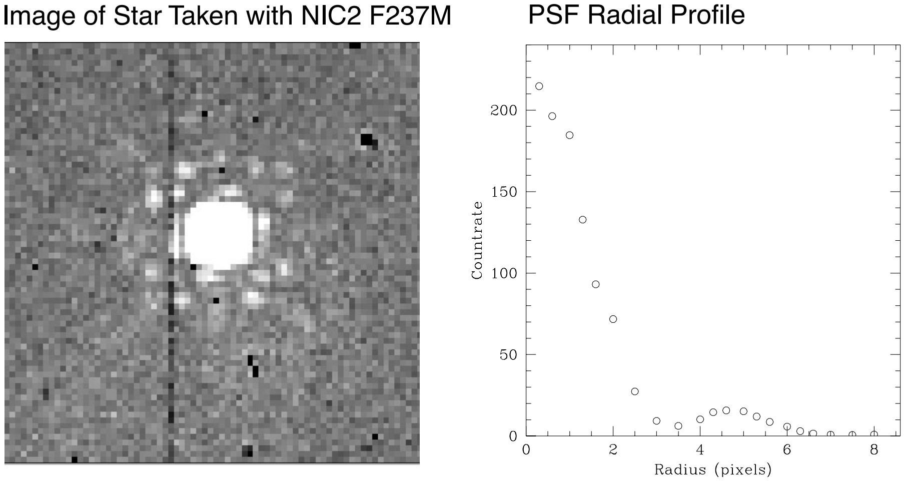


Image of Star Taken with NIC2 F237M

PSF Radial Profile

5.5.1 Impact of Instrumental Effects on PSF Subtraction

There are a number of factors which affect the NICMOS PSF and thus can impact the results of image analysis relying on PSF subtraction. The factors are: focus variations due to OTA breathing, cold mask irregular motion ("wiggling") on an orbital time scale, PSF color dependence.

The limits to the accuracy of PSF subtraction imposed by these factors have been assessed using model PSFs generated by **TinyTim** software and **synphot** generated blackbody spectra. The results for Camera 2 are given in table 5.6 below. The mean and standard deviation of the PSF subtraction residuals are given a percentage of the nominal PSF pixel values. The standard deviation can be interpreted as the average relative spatial noise expressed as a percent of the PSF pixel value at any given distance from the PSF center. The mean represents a systemic component in the PSF residuals.

Table 5.6: Effect of focus variation, cold mask wiggling, and PSF color dependence on PSF subtraction.

effect of:	mean sigma F187W	mean sigma F187N	mean sigma F160W	mean sigma F187N	mean sigma F110W	mean sigma F187W
focus variation	0.62% 6.81%	1.86% 17.29%				
cold mask wiggling			0.10% 14.38%	3.15% 37.67%		
color dependence (8000K-3000K)					23.50% 8.91%	0.58% 1.57%
color dependence (8000K-6000K)					-1.52% 1.32%	
color dependence (6000K-3000K)					25.31% 7.32%	

The effects of both cold mask wiggling and focus breathing introduce errors in a PSF-subtracted image, well above 20% of the PSF signal in a narrow band filter, with a spatial scale of a few pixels. The main effect of PSF color dependence is adding a systemic component to the PSF subtracted image. The effect is quite large if the color of the PSF used for subtraction is very different from the image PSF color. It is pronounced in blue, wide filters, like F110W, while in filters like F187W or redder/narrower it is essentially negligible. The best way to cope with this effect is to use PSFs with colors well matched to the sources.

PSF subtraction with NICMOS Camera 3 is extremely difficult due to the large pixel scale which severely undersamples the point spread function, and because of the strong intrapixel sensitivity variations (see section 5.3.3) which affect the structure of point source images.

Figure 5.4: Effect of cold mask wiggling on PSF subtraction in broad band and narrow band filters. Left panels: normalized counts (log scale) in the central column of the two **TinyTim** PSFs with 0.005 difference of the cold mask offset (thin lines). The thick line is the difference between the two. Right panels: absolute value of counts difference as a percent of the counts for the PSF at the "nominal" cold mask offset.

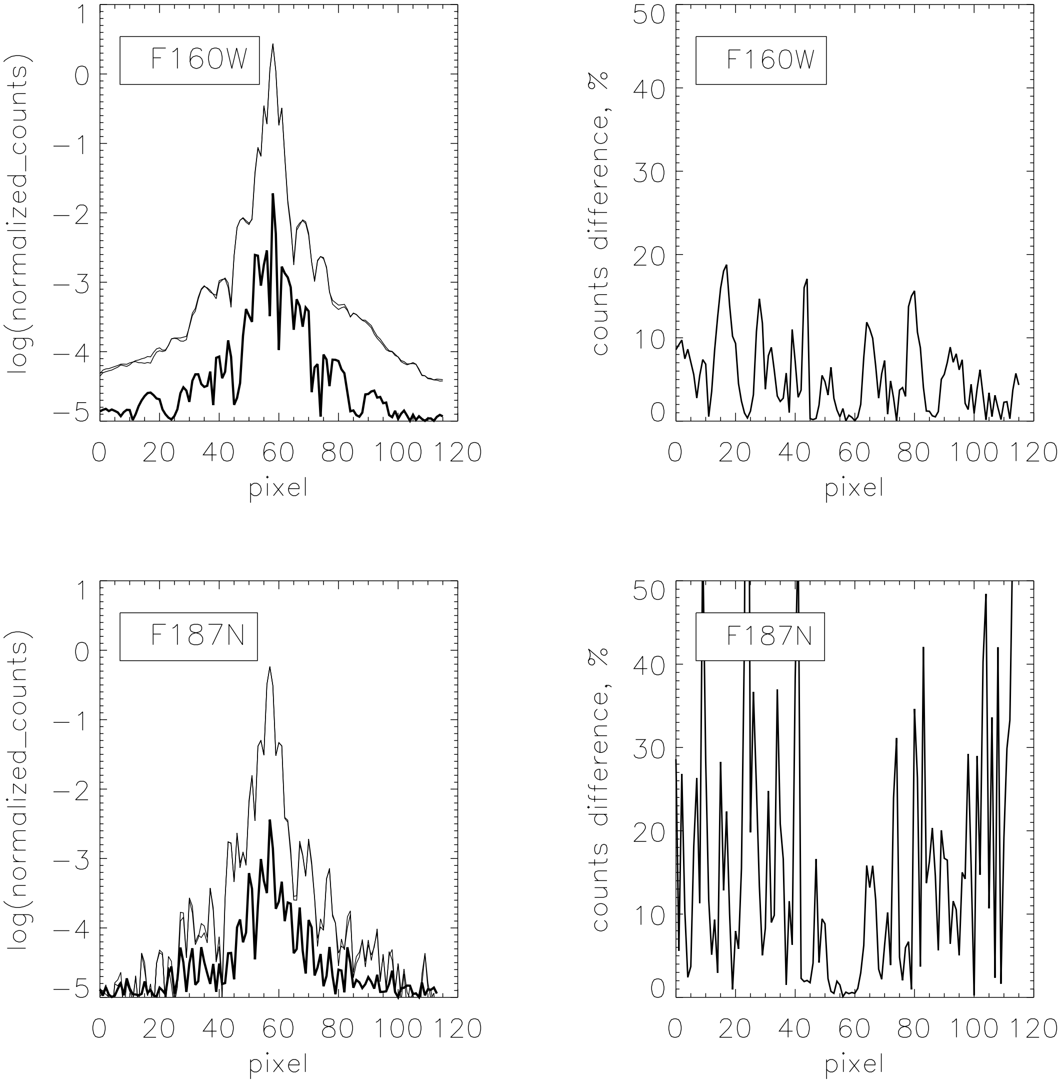


Figure 5.5: Illustration of PSF color dependence in different filters. In F110W, the 8000 K blackbody spectrum has a very different slope from that of the 3000 K blackbody. Therefore the PSF resulting from images of point sources with these temperatures will be very different from one another. The F187W filter, however, samples the Rayleigh-Jeans regime for both temperatures, resulting in spectra with about the same slope, so the difference between the 8000 K PSF and 3000 K PSF will be very small.

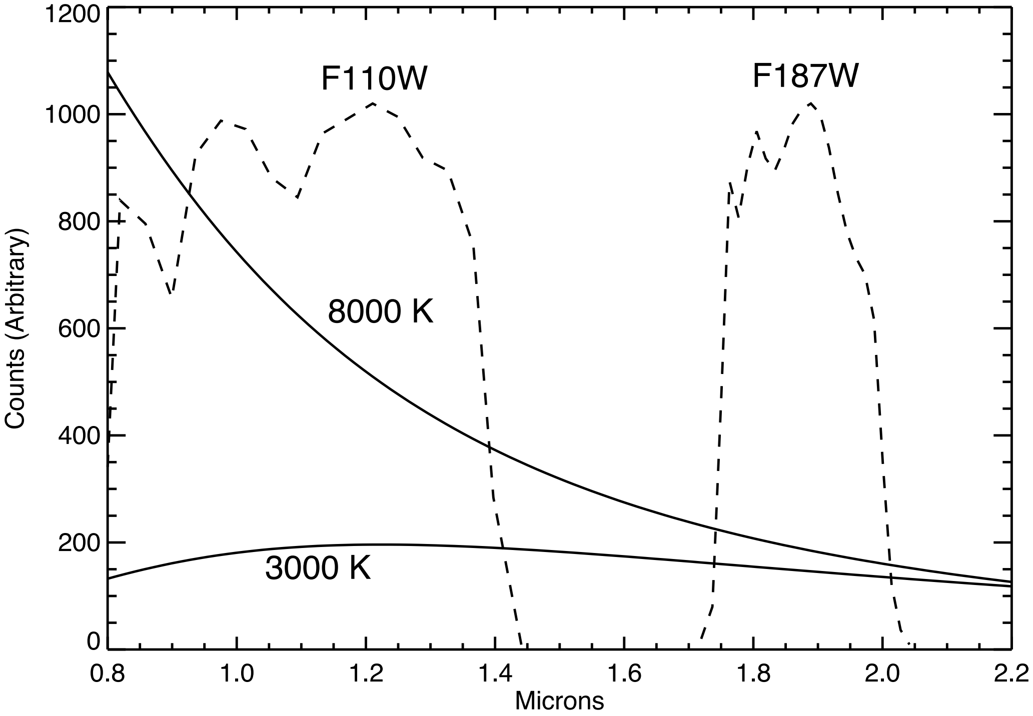
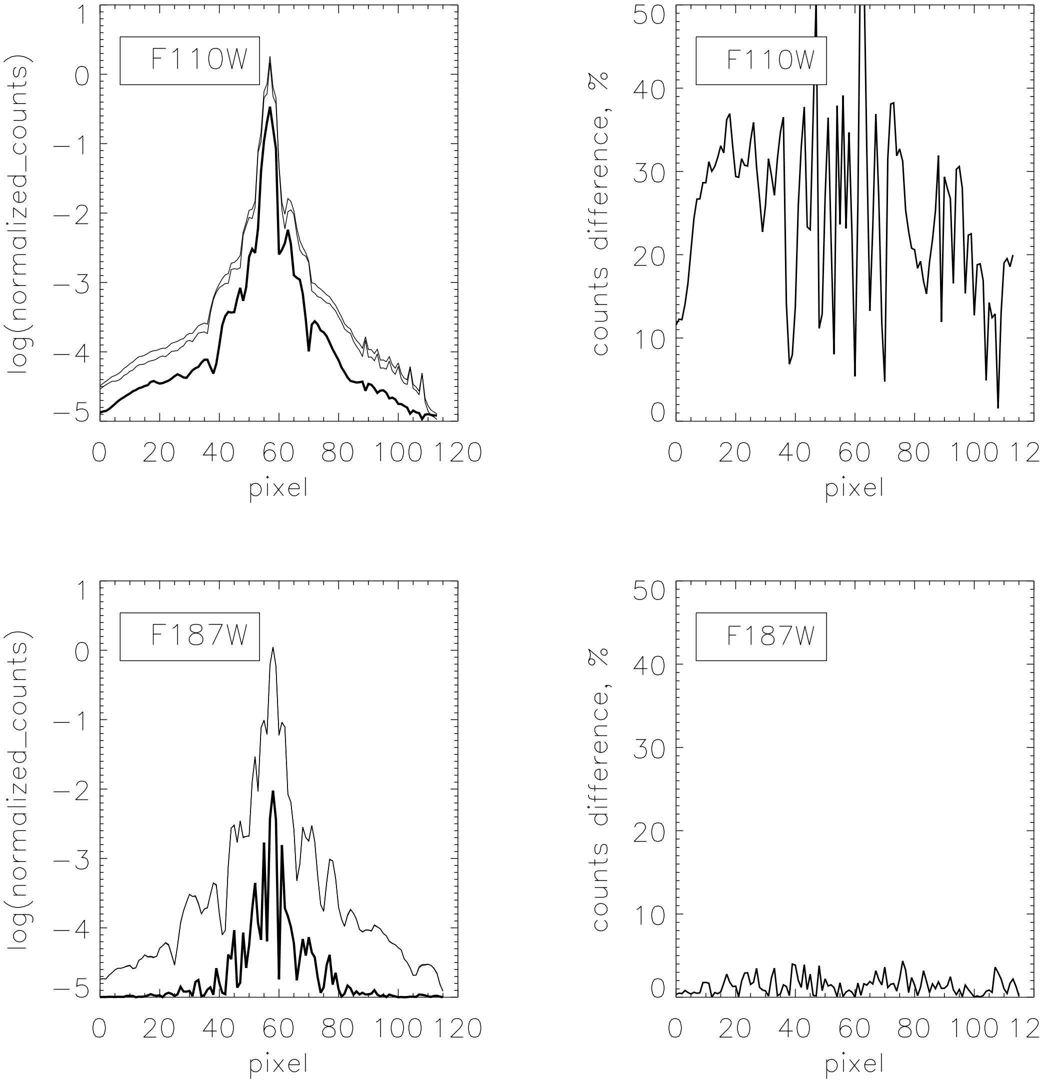


Figure 5.6: Effect of PSF color dependence on PSF subtraction in different filters. Left panels: normalized counts (log scale) in the central column of **TinyTim** 8000 K and 3000 K blackbody PSFs (thin lines). The thick line is the difference between the two. Right panels: absolute value of counts difference as a percent of counts for the 8000 K PSF.



5.6 Coronagraphic Reductions

5.6.1 Data Products and File Structures

NICMOS coronagraphic observations consist of an acquisition image and science observations. The acquisition could be an onboard acquisition performed by the NICMOS flight software (FSW) during the same visit as the science observations. Or, a real time acquisition may have been performed by sending the acquisition image(s) to the ground for analysis by the Observation Support System (OSS) and Post-Observation Data Processing System (PODPS) Unified System (OPUS) personnel during the first visit of a two visit observation set. The accompanying science observations would have been obtained during the second visit. The ACCUM, BRIGHTOBJ, and MULTIACCUM observing modes have all been used for acquisitions and for the science observations. Each type of data requires different calibration steps and dark reference files.

5.6.2 Coronagraphic Acquisitions

Coronagraphic imaging requires an acquisition sequence at the beginning of the visit to center the target in the coronagraphic hole. The size of the coronagraphic hole is smaller than typical HST blind-pointing errors. The procedure for a coronagraphic acquisition is to first image the target in Camera 2 using blind-pointing and then use either an onboard (using the NIC2-ACQ aperture), reuse target offset, or interactive acquisition to acquire the target. A telescope slew is calculated and commanded to move the position of the hole over the image of the target.

During the Second Servicing Mission (SM2) Science Mission Orbital Verification (SMOV), it was determined that decentering a point source by a small amount, x=-0.75, y=-0.25 pixels, from the center of the hole reduced the background intensity (program ID: 7052, see NICMOS IDT report "Results from SMOV 7052 NICMOS Coronagraphic Performance Verification"). Based on the SMOV results, this offset was implemented in the NICMOS ACQ flight software for use in Cycle 7 science observations.

The target is positioned on the NIC2-CORON aperture, which has the aperture fiducial point at the image position of the coronagraphic hole on the detector. The science exposures are then specified using any of the NICMOS observing modes and any of the NIC2 filters.

The various modes of NICMOS coronagraphic acquisition and the processing required to analyze the resulting data are described in detail in a series of Instrument Science Reports (ISRs) by Schultz et al., which are available from the STScI WWW pages. Users analyzing coronagraphic data, or planning future coronagraphic observations, should carefully read NICMOS ISR-031, ISR-98-012, ISR-98-019, and ISR-99-006. Some important highlights for understanding and reducing coronagraphic data are summarized here, but the reader should consult the ISRs for more detailed discussions.

Onboard Acquisition

In Mode-2 acquisition (see NICMOS ISR-98-012, Schultz et al.), the onboard NICMOS flight software (FSW) will locate the position of the coronagraphic hole and target. It will use this information to calculate an offset to slew the telescope to position the image of the hole over the position of the target. The location of the coronagraphic hole is determined from pointed flat field observations. Two short ACCUM F160W filter exposures (7.514 seconds each) with calibration Lamp 1 on (flat field) and two identical exposures with the lamp off (background) are obtained before the acquisition images. The telescope is NOT slewed to move the target out of the FOV during the lamp-on and background observations. The image of the target will be superimposed on these images.

The flight software combines the two background and two flat field images, subtracts the background image from the flat field image, and determines the position of the hole. These temporary difference images are not saved. The location of the hole is temporarily stored onboard, but it is not included in the science telemetry. The target position and the slew are stored within the embedded engineering attached to the data set. This information is extracted from the engineering telemetry and written to the `ipppssoot_spt` image. The two flat field images are written to image `ipppssoot_rwf.fits` and the two background images are written to image `ipppssoot_rwb.fits`.

An error in the Flight Software code prior to June 28, 1998 resulted in errors in the determination of the image position of the coronagraphic hole. Usually, this error was at the sub-pixel level and occasionally was off by a pixel. The decentered imaging affected the coronagraphic performance. It was corrected after this date.

The acquisition aperture is a square area on the detector, a 128 x 128 pixel aperture (center at 157, 128) of size 9.6 x 9.6 arcseconds. Two ACCUM images of equal exposure are obtained, and are saved together in a single fits file. A summary of the files produced in an ACQ data set is presented in table 5.7.

Table 5.7: NICMOS ACQ data image extensions

Extension	Image Contents
ipppssoot_raw	Raw target data
ipppssoot_rwb	Raw background data
ipppssoot_rwf	Raw flat field data
ipppssoot_spt	Target SHP and UDL information
ipppssoot_spb	Background SHP and UDL information
ipppssoot_spf	Flat field SHP and UDL information
ipppssoot_cal	Calibrated target data
ipppssoot_clb	Calibrated background data
ipppssoot_clf	Calibrated flat field data
ipppssoot_trl	OPUS processing trailer file
ipppssoot_pdq	OPUS Processing Data Quality file

During Cycle 7 and 7.5, the exptime keyword for NICMOS ACQs was incorrectly populated. A correction was installed in OPUS 9.4, on October 19, 1998.

Reuse Target Offset (RTO) and Interactive Acquisitions

A variation of the Reuse Target Offset (RTO) capability was sometimes used to acquire and position bright targets into the coronagraphic hole. This is known as Mode-1 acquisition (see Schultz et al., NICMOS ISR-98-019). Any target that would saturate the detector in the shortest possible Mode-2 ACQ ACCUM exposure time, 0.228 seconds, was considered to be a bright target. The following discussion describes the necessary steps for a Mode-1 Reuse Target Offset (RTO) acquisition. The RTO acquisitions were performed during the first visit and the science observations were obtained in a following second visit.

Images of the target and coronagraphic hole were obtained a few orbits in advance of the coronagraphic observations, and sent to the ground for analysis (RT ANALYSIS). The same roll of the spacecraft was used for both the acquisition and science visits. Usually, the same dominant and subdominant guide stars were used for both the acquisition and science observations. However, in a few instances, this was not the case; after the slew was performed, the target was found outside of the hole, either on the edge of the hole or far from the hole. The coronagraphic observer as well as future Archive users are advised to check the OPUS PDQ files for suspect observations.

The background and flat field observations were usually offset as much as 18-25 arcseconds from the target position to avoid the diffraction spike from the image of an overexposed target crossing the coronagraphic hole and introducing errors in the measured position of the coronagraphic hole. OPUS staff assisted the PI in identifying the target, centroiding, and determining offsets. A record of the acquisition was written to the PDQ file.

5.6.3 Positions of the Hole and Target

The location of the target and the slew are saved in the embedded engineering data attached to the science observations. These values are recovered from the embedded engineering data and written to keywords (NXCENT, NYCENT, NOFFSETX, NOFFSETY) in the SPT file. The first observation following the NICMOS ACQ will contain the updated values. The target location and slew values of the most recent target acquisition will be reflected in following observations until the next acquisition executes, or until the values are re-initialized to zero. These values are not initialized during normal operation of NICMOS.

As presented, the keyword values are scaled ENGINEERING units, and not in pixels or arcseconds. They must be converted from ENGINEERING units to DETECTOR and IMAGE coordinates.[5]

Target Position in Detector Coordinates

The target position coordinates can be converted into DETECTOR coordinates by dividing by 256. However, the slew values are written in 2's compliment. A value less than 32757 is positive, while a value larger than 32757 is negative. Slew values need to be divided by 128 during the conversion into DETECTOR coordinates. For example, the following target and slew values were obtained from an SPT file and converted to DETECTOR coordinates.

[5] NICMOS Instrument Science Report, NICMOS-ISR-98-012.

```
> hedit n4q832nrq_spt.fits[1] NXCENT,NYCENT,NOFFSETX,\
>>> NOFFSETY .
n4q832nrq_spt.fits[1],NXCENT = 31418
n4q832nrq_spt.fits[1],NYCENT = 20655
n4q832nrq_spt.fits[1],NOFFSETX = 10101
n4q832nrq_spt.fits[1],NOFFSETY = 52259

NXCENT = 31418.0 / 256.0 = 122.726
NYCENT = 20655.0 / 256.0 = 80.683
NOFFSETX = 10101.0 / 128.0 = 78.914
NOFFSETY = (52259.0 - 65535.0) / 128.0 = -103.718
```

The position of the coronagraphic hole can then be inferred by subtracting the offset slew and the coronagraphic hole offset from the position of the target.

```
NXHOLE = 122.726 - 78.914 - 0.25 = 43.562
NYHOLE = 80.683 + 103.718 - 0.75 = 183.651
```

A fix was included in OPUS 9.2, which was installed July 16, 1998. The slew and target position keywords NXCENTP, NYCENTP, NOFFSTXP, and NOFFSTYP were added to the SPT file. These keyword values are in detector coordinates.

The IMAGE coordinate system for NICMOS Cameras 1, 2, 3 have been defined such that the origin will be in the lower left hand corner when displayed, while DETECTOR coordinates are defined by the readout directions for each camera. Any NICMOS camera image, when displayed using the IRAF **display** command, will be displayed relative to the HST field of view as depicted in the NICMOS Instrument Handbook. The conversion from DETECTOR coordinates to IMAGE coordinates is performed during OPUS pipeline processing. For Camera 2, the OPUS +x direction is detector -y direction, and correspondingly, the OPUS +y direction is detector -x direction. Care must be exercised when converting from one coordinate system to the other.

Measuring the Hole Position

The coronagraphic hole location can be determined from off-line processed ACQ images and should be very close to the inferred location of the hole determined by subtracting the FSW slew from the FSW target position. However, the hole pattern is not symmetric about the low scatter point (pixel with least counts) of the OPUS pipeline processed coronagraphic hole image due to the impression of the flat field on top of the image. The onboard ACQ background and flat field images need to be reduced in a similar manner as performed by the FSW to achieve a meaningful comparison. The coronagraphic acquisition n4q832npq will be used as an example to demonstrate off-line processing.

The background image (n4q832npq_rwb.fits) must be subtracted from the image of the hole (n4q832npq_rwf.fits), and the resulting image flat fielded using a preflight flat or SMOV flat with the hole at a different position. In this example, the pre-flight flat field h1s1337dn_flt.fits was used to process the hole image. The task **imcalc** is used to combine the pair of background and hole images, taking the minimum value at each pixel in order to eliminate cosmic rays.

```
> imcalc n4q832npq_rwb01.fits[1],n4q832npq_rwb02.fits[1] \
>>> min_bac "min(im1,im2)"


> imcalc n4q832npq_rwf01.fits[1],n4q832npq_rwf02.fits[1] \
>>> min_fla "min(im1,im2)"


> imarith min_fla - min_bac min_flat_bac

> imarith min_flat_bac * nref$h1s1337dn_flt.fits[1] \
>>> pro_flat_bac
```

One can then invert the flat fielded hole image to make the hole positive, and then use the IRAF task **center** to determine the centroid of the reversed hole image.

The position of the coronagraphic hole measured from the pipeline processed image is (73.384, 213.082), while the measured position of the hole from the off-line processed image is (73.273, 213.163). The difference is ~0.1 pixel in the x- and y-positions. This measured offset in hole position is most probably due to the different calibrations performed on the ACQ images. The pipeline processed ACCUM images are not dark or background corrected. The location of the coronagraphic hole in the pipeline on-orbit flat was patched and could contribute to error in determining the position of the hole.

For comparison with the FSW determined position of the hole, the IRAF positions of the hole need to be converted into detector coordinates by subtracting them from 256.5. For example:

```
Hole:

NYHOLE = 256.5 - 73.273 = 183.227

NXHOLE = 256.5 - 213.163 = 43.337
```

In this example, for back-to-back orbit ACQ images n4q832npq and n4q833nzq (obtained prior to June 28, 1998), the difference between the two positions derived for the hole (IRAF and FSW) in detector coordinates is ~0.9 pixels in y and ~0.3 pixels in x. This is much larger than can be explained by summing the errors of the target and hole positions in quadrature, and is due to a bug in the early version of the FSW, which introduced errors in the derived hole positions.

Measuring the Target Position

The IRAF task **center** can be used to determine the centroid of the target in the NIC2 field of view. For comparison with the FSW positions of the target, the centroid values need to be converted into detector coordinates by subtracting them from 256.5. For example:

```
Star:

NXCENTER = 256.5 - 133.725 = 122.775

NYCENTER = 256.5 - 175.708 = 80.792
```

In this example, the target position we have measured agrees quite well with that determined by the FSW. The slight differences are most probably due to the different algorithms used to determine the centroids.

5.6.4 Recalibrating Coronagraphic Images

The user may wish to recalibrate coronagraphic data using the best reference files currently available (see section 3.5). Unfortunately, at this time there are no standard dark reference files available from the calibration database for ACCUM and BRIGHTOBJ mode observations (see section 4.1.3 for a discussion). Some coronagraphic observers obtained their own on-orbit ACCUM mode darks, which are available from the HST Archive and can be used for calibrating the science data. If you have questions about calibrating coronagraphic ACCUM images, please contact the STScI help desk (help@stsci.edu).

Any given on-orbit flatfield observation with NIC2 will include the coronagraphic hole. Because the hole moved with time, it is unlikely that the hole will be in the same location for a given observation as it was in a flatfield taken at a different time. This can have particularly catastrophic consequences for coronagraphic data. It is therefore important to use a flatfield where the hole is in a substantially different location (e.g., a pre-flight flat field from thermal vacuum testing, or from early in SMOV when the hole location was very different), or a flat where the hole has been “patched.” The current generation of NIC2 flat field reference files available from the STScI calibration database have the hole patched. This should be suitable for most purposes, although the patches are not perfect, and it is possible that small mismatches may still affect coronagraphic science. Also, the preflight flat fields were obtained at a different instrument temperature, and because the flat field structure is known to vary with temperature the preflight flats will not provide a perfect match to on-orbit data. A future release of NIC2 flat fields should include better hole patches. For reference, table 5.8 lists the preflight Camera 2 flat fields.

Table 5.8: NICMOS Camera 2 preflight calibration flat field files

filter	flat field	filter	flat field
F110W	h1s1337cn	F207M	h1s1337mn
F160W	h1s1337dn	F212N	h1s1337nn
F165M	h1s1337en	F215N	h1s1337on
F171M	h1s1337fn	F216N	h1s1337pn
F180M	h1s1337gn	F222M	h1s1337qn
F187N	h1s1337hn	F237M	h1s1337rn
F187W	h1s1337in	POLOL	h1s13380n
F190N	h1s1337jn	POL120L	h1s1337sn
F204M	h1s1337kn	POL240L	h1s1337tn
F205W	h1s1337ln		

Contemporary "Hole" Flats

Calibration using contemporary flat fields would remove the very strong hole-edge gradient resulting from calibration with a non-contemporaneous hole image or with a patched flat. The F160W filter acquisition lamp on/off paired images obtained as part of the Mode-2 target acquisition process can be used to create a flat field reference file. The hole position in these images is at the same location as the hole position in the associated coronagraphic observations. Reference files created from these images would be appropriate for regions close to the coronagraphic hole. The S/N for these F160W filter reference files would be S/N ~100.

Note: the recommended standard calibration reference files have high S/N (typically ~ 1200) and should be used for regions far from the coronagraphic hole. The Mode-2 acquisition F160W filter lamp and background images will usually contain an overexposed image of the target.

For completeness, the following processing steps to create a contemporary flat field from the ACQ images are listed to assist the user.

```
ni> hedit n4xj13jwq_rwb.fits[0] flatcorr OMIT ver-

ni> hedit n4xj13jwq_rwb.fits[0] unitcorr PERFORM ver-

ni> hedit n4xj13jwq_rwb.fits[0] photcalc OMIT ver-


ni> hedit n4xj13jwq_rwf.fits[0] flatcorr OMIT ver-

ni> hedit n4xj13jwq_rwf.fits[0] unitcorr PERFORM ver-

ni> hedit n4xj13jwq_rwf.fits[0] photcalc OMIT ver-


ni> calnica n4xj13jwq_rwb.fits "" > n4xj13jwq_btrl.log

ni> calnica n4xj13jwq_rwf.fits "" > n4xj13jwq_ftrl.log


ni> mssplit n4xj13jwq_clb.fits

ni> mssplit n4xj13jwq_clf.fits


ni> mscombine n4xj13jwq_clb01.fits,n4xj13jwq_clb02.fits \
>>> n4xj13_bck

ni> mscombine n4xj13jwq_clf01.fits,n4xj13jwq_clf02.fits \
>>> n4xj13_lamp


ni> msarith n4xj13_lamp.fits - n4xj13_bck.fits \
>>> n4xj13_lamp_bck


ni> msstat n4xj13_lamp_bck.fits[sci,1][1:256,36:256]

ni> msarith n4xj13_lamp_bck.fits / 1169.64 norm_lamp_bck


ni> msarith 1.0 / norm_lamp_bck n4xj13_f160w_flt

ni> hedit n4xj13_f160w_flt.fits[0] PEDIGREE \
>>> "Contemporary 31/08/98" add+
```

Recalibrating the ACQ ACCUM Images

Some observers have expressed the desire to obtain photometry from the ACQ images of their targets. These images are not dark corrected and will need to be recalibrated. As discussed in section 4.1.3, however, at the present time there are no standard dark reference files for ACCUM mode images. Appropriate DARK exposures matching the ACQ image exposure times may be available in the HST Archive. If you are a NICMOS GO, please speak with your contact scientist (CS) about dark subtraction for ACCUM images, or contact help@stsci.edu.

We note here that due to a software bug, the exposure time stated in the header keyword `EXPTIME` is in error for ACQ data taken prior to October 19, 1998. The commanded exposure time listed in the SHP and UDL data file (`_spt.fits`) should be used for photometry. For example:

```
> hedit n4xj13jwq_raw.fits[0] EXPTIME .
n4xj13jwq_raw.fits[0],EXPTIME = 0.625648


> hedit n4xj13jwq_spt.fits[0] CMD_EXP .
n4xj13jwq_spt.fits[0],CMD_EXP = 0.255647987127
```

The SAMPTIME keywords have to be changed in both SCIENCE groups of the raw, uncalibrated imageset (`_raw.fits`) before **calnica** processing.

5.6.5 Reducing and Co-adding Coronagraphic Images

The OPUS pipeline will combine multiple coronagraphic exposures using **calnicb**. This can cause difficulties, however, for coronagraphic data analysis. If the individual images that form a mosaic require registration, some smoothing will be introduced by the bi-linear interpolation that is used by **calnicb** to shift the images. Even if the images are well registered to begin with, slight differences in the diffraction pattern due to movement of the coronagraphic hole may cause the **calnicb** cross-correlation routine to compute non-zero offsets between the various images. This will lead to misregistration and again, some smoothing. One way around this problem is to force **calnicb** not to shift any of the images before averaging them together. This can be accomplished by adding columns of `XOFFSET` and `YOFFSET` values to the input association table (`_asn.fits`) used by **calnicb** and setting their data values to zero. However, in general, PSF subtraction is best performed with individual images and not with the mosaic image.

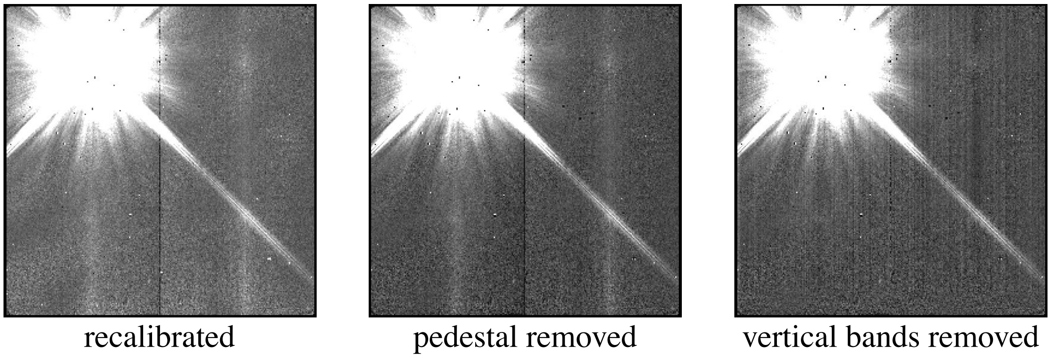
Pedestal Removal

The residual bias or “pedestal” effect was described in section 4.1.2, and should be removed if possible before analyzing coronagraphic data. The **pedsky** task described in section 4.1.5 may not be suitable for coronagraphic images where a bright object dominates the field of view. In this case, the **pedsub** task may be more appropriate. This method has been used successfully with coronagraphic images from the SMOV calibration program 7052 (1997 July 23). The **pedsub** task was run with the parameters `filter=mask` (i.e., applies unsharp-masking filter to remove low spatial frequency information) and `doquadeq=no` (i.e., do not force quandrant boundaries to be continuous). The images of the stellar PSF in the coronagraphic images of the 7052 data fill the quadrant with the coronagraphic hole and spill over into adjacent quadrants. This constrains the determination of the pedestal contribution in that quadrant. A mask file was created to flag bright pixels (threshold limit= 4.0 cts/sec); the flagged pixels were not used for estimating the pedestal contribution. This mask was added to the mask of known bad pixels included in the DQ image extension, and the **pedsub** parameters `dqon` and `dqpar` were used to tell the task which mask values correspond to pixels that should be ignored when fitting the pedestal.

Removing Other Image Artifacts

The identification and removal of bad pixels, “grot,” and other NICMOS data anomalies are discussed in chapter 4. For coronagraphic data, the presence of a bright source commonly induces the vertical streaks known as the “Mr. Staypuft” effect (see section 4.6.3). Because of the bright coronagraphic target, it is not possible to fit simple medians to columns along the entire y-axis length of the image in order to measure and subtract the Mr. Staypuft streaks. It may be possible to remove (or at least reduce) the streaks using a correction image derived from the bottom rows of the image only, far from the coronagraphic target. The first 19 rows are usually the only area of the image that is not intersected by the bright diagonal diffraction spike emanating from the hole. You may use **blkavg** to average the first 19 rows, and then **blkrep** to “stretch” this row average to form a 2-dimensional image to be subtracted from your data. Care must be taken to ensure that bad pixels (hot or cold pixels, or “grot”) do not bias the row average: if necessary, interpolate over bad pixels using **fixpix**.

Figure 5.7: Coronagraphic image, F110W filter. Star centered in hole with image recalibrated (left), pedestal removed (middle), and with "Mr. Staypuft" striping partially corrected (right). Images are displayed with the same stretch.



In figure 5.7 above, the correction for the Mr. Staypuft bands is not perfect, and shows the limitations of doing this. The amplitude may modulate from one quadrant to another for electronic reasons, and flatfielding also introduces variations along the column so that a constant correction derived from the bottom rows may not be appropriate. Nevertheless, the process results in first-order cosmetic improvement, and may be worthwhile.

Persistent afterimages from previous exposures of the target star (see section 4.6.1) can also be a problem for coronagraphic images, and users should be aware of this effect.

PSF Subtraction

The light distribution within the PSF and the pattern of light scattered about the hole can change significantly on both short and long time scales, from orbit to orbit and over the lifetime of NICMOS (see section 5.5 for a general discussion of focus and PSF issues). These changes may be attributed to several factors, including OTA focus variations, thermally induced motions in NICMOS fore-optics, possible changes in the position of the cold mask, and coronagraphic hole motion. All of these changes are related to either the thermal input to the telescope, changes in Sun angle (attitude) and spacecraft roll, or to the dewar thermal short and subsequent cryogen depletion. The PSF also varies as a function of position across the field of view, which may affect the quality of PSF subtractions (see figure 5.8). In addition, the HST focus is known to oscillate with a period of one HST orbit. Changes in the focus are attributed to thermal contraction/expansion of the optical telescope assembly (OTA) resulting from the telescope warm up and cool down during an orbital period.[6] These short term focus variations are usually referred to as "OTA breathing", "HST breathing", "focus breathing", or simply "breathing".

[6] NICMOS Instrument Science Report, NICMOS ISR-98-015

During Cycle 7 and 7.5, the NICMOS dewar anomaly caused the coronagraphic hole to migrate to different locations on the detector. The position of the hole on the detector had been observed to move as much as ~0.25 pixel in three orbits. The movement of the hole was found not to be uni-directional, but rather, the hole “jitters” back and forth along an X-Y diagonal by as much as ± 0.5 pixel.[7]

The direct subtraction of two unregistered coronagraphic images that were not obtained back-to-back in the same orbit with a change in roll can yield large residuals. Coronagraphic images need to be subpixel shifted and/or convolved with a Gaussian function to match the observed PSF before subtraction.

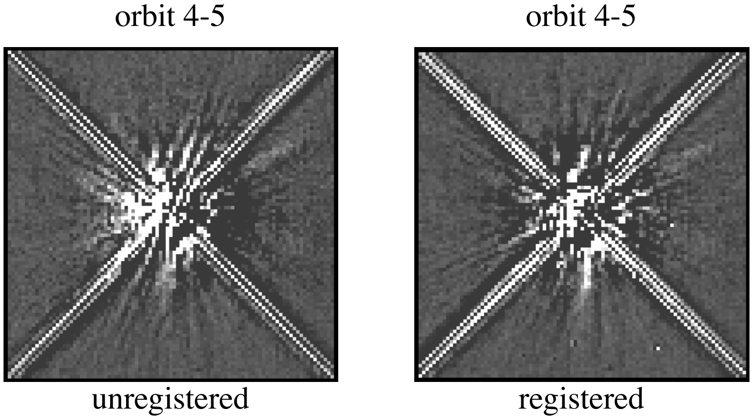
The following discussion applies both subpixel shifting and Gaussian convolution reduction techniques to two F110W filter images from the NICMOS calibration program 7052. The data used in the examples below are n45j22lam and n45j23mym. The data were obtained in consecutive orbits with a spacecraft roll of 36° between orbits.

Subpixel Shifting

The images to be subtracted must be registered to sub-pixel accuracy in order to achieve good results. The IRAF task **xregister** provides one way to determine the offsets between the images, and to shift one into alignment with the other. For the example considered here, a 995 pixel region ([27:121,162:256]) around the hole was used as the area for cross-correlation, and a shift of dx = -0.286, dy = 0.136 pixels was measured between the images. The example shown in figure 5.8 and figure 5.9 displays the subtraction of the two images, using both unregistered (left) and registered (right) data. The subtraction using the registered images exhibits a more symmetrical residual light pattern about the hole. Occasionally, cross correlation does not yield the best measure of the image offsets. You may find that “trial-and-error” shifting by various fractional pixel amounts, minimizing the subtraction residuals “by eye,” can yield the best results.

[7] NICMOS Instrument Science Report, NICMOS-ISR-98-012.

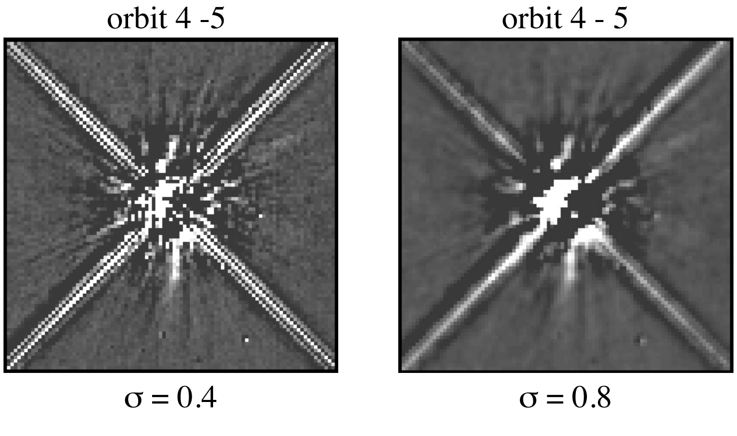
Figure 5.8: PSF Subtraction. F110W filter images obtained in back-to-back orbits with a roll of the spacecraft between orbits. Direct subtraction of images (left) and a subtraction with the second image shifted to match the first image (right).



Gaussian Convolution

The interpolation introduced by sub-pixel shifting has the effect of smoothing the image slightly. Subtracting the blurred image from the unblurred image has the effect of edge enhancement. One way to suppress high frequency variations in the images is to convolve one or both by a Gaussian filter. This may help reduce residuals in the subtracted image. The IRAF task **gauss** may be used to smooth the images. The resulting images for two different convolutions, with $\sigma = 0.4$ and $\sigma = 0.8$, are presented in figure 5.9.

Figure 5.9: PSF Subtraction. F110W filter images obtained in back-to-back orbits with a roll of the spacecraft between orbits. Images convolved with a Gaussian function, $\sigma = 0.4$ (left) and $\sigma = 0.8$ (right) and subtracted.



In this example, the $\sigma = 0.4$ convolution results in a slight improvement over the subtractions shown in figure 5.8. Possibly, this choice is undersmoothing the data. The $\sigma = 0.8$ convolution more closely matches the low frequency components. Determining a suitable degree of smoothing will require experimentation by the user.

5.7 Analysis of Polarization Images

5.7.1 Introduction

The filter wheels of cameras NIC1 and NIC2 each contain three polarizing filters with unique polarizing efficiencies and position angle offsets. The original design specified that the position angle of the primary axis of each polarizer (as projected onto the detector) be offset by 120° from its neighbor, and that the polarizers have identical efficiencies. While this clean concept was not strictly achieved in NICMOS, the reduction techniques described below permit accurate polarimetry using both cameras over their full fields of view.

A complete set of polarimetric observations will contain images obtained in all three polarizers of the selected wavelength range. We assume that each image has been processed through **calnica** and **calnicb** to produce a fully reduced and (if necessary) mosaiced image in each of the three filters, with the data corrected for saturation and cosmic rays and converted to flux density using the appropriate photometric calibration constants for the polarizers.

To generate Stokes parameters, the relative differences in flux between images in the different polarizing filters are used. Where the signal level is very faint, and the signal-to-noise ratio is very low, the differences may be very large but dominated by noise. If you attempt to calculate the Stokes parameters using such data, you will likely obtain large and entirely spurious polarizations. Therefore, it is not advisable to use low signal-to-noise data to calculate polarization. To avoid this problem, it is suggested to estimate the noise in an area of the image free of sources, and then set a threshold at a value of order five to ten times this noise level. Using the IRAF task **imreplace**, all pixels with signals below this threshold should be set to some arbitrary value, probably close to the measured noise level. This action will cause all areas of the image where the signal level is very faint to show zero polarization. To further increase signal-to-noise, bin the data in each of the three images before computing the Stokes parameters. Once the parameters have been derived, clipping the Q and U images at ±1 and the polarization intensity image at values < 0 or >1 will also help in increasing the signal-to-noise.

"Analysis of Polarized Light with NICMOS" by Hines, Schmidt & Schneider (2000, PASP, 112, 983) is a highly valuable reference discussing NICMOS polarimetric measurments and data analysis.

5.7.2 Theory

In order to reduce data obtained with a set of polarizers, three quantities are needed. They are the throughput for unpolarized light, the efficiency of the element as a polarizer, and the orientation of the polarizer. These quantities can be expressed in a polarization reduction algorithm to form a solution containing the polarization characteristics of the incoming beam (i.e., the Stokes parameters I, Q, and U).

The general form of the equation for polarimetric data reduction is expressed as

$$I_k' = A_k I + \varepsilon_k (B_k Q + C_k U)$$

where I_k' is the emerging light intensity from the kth polarizer, A_k, B_k and C_k are the transmission coefficients, and ε_k is the polarizing efficiency.[8] This linear equation captures the observed signal from a polarized source of intensity I and linear Stokes parameters Q and U, which describe the state of polarization for the target object. The above equation reduces to a set of three equations with three unknowns. The solution results in the Stokes parameters for the incoming light.

For NICMOS the transmission coefficients are[9]

$$A_k = \frac{(t_k)}{2}(1 + l_k)\ ,\ \ B_k = \frac{1 - l_k}{1 + l_k}\cos 2\phi_k, \quad C_k = \frac{1 - l_k}{1 + l_k}\sin 2\phi_k$$

where ϕ_k is the position angle of the kth polarizer relative to the NICMOS entrance aperture, t_k is the fraction of light transmitted for a 100% polarized input aligned with the polarizer's axis, and l_k is the fraction transmitted when the incoming light is perpendicular to the axis of the polarizer.

***For NICMOS, the observed signal from a polarized source of total intensity I contains an added term in the transmission coefficients, namely* $\mathbf{0.5(1 + l_k)}$.**

Table 5.9 below presents the properties of the individual polarizers.

[8] Polarizer efficiency is defined as $\varepsilon = (S_{par} - S_{perp}) / \langle S_{par} + S_{perp} \rangle$ where S_{par} and S_{perp} are the respective measured signals for a polarizer oriented parallel and perpendicular to the axis of a fully polarized beam.

[9] For further detailed information on the derivation of the coefficient matrices, see Hines, D.C. “Imaging Polarimetry with NICMOS”, VLT Conference 1998.

Table 5.9: NIC1 & NIC2 Polarizer Properties

	NIC 1					NIC 2			
Filter	φ_k	ε_k	t_k	l_k	**Filter**	φ_k	ε_k	t_k	l_k
POL0S	1.42	0.9717	0.7760	0.0144	POL0L	8.84	0.7313	0.8981	0.1552
POL120S	116.30	0.4771	0.5946	0.3540	POL120L	131.42	0.6288	0.8551	0.2279
POL240S	258.72	0.7682	0.7169	0.1311	POL240L	248.18	0.8738	0.9667	0.0673

The resulting coefficient matrices become[10]

$$M_{NIC1} = \begin{bmatrix} 0.3936 & 0.3820 & 0.0189 \\ 0.4025 & -0.1166 & -0.1526 \\ 0.4054 & -0.2876 & 0.1195 \end{bmatrix}, \quad M_{NIC2} = \begin{bmatrix} 0.5187 & 0.3614 & 0.1152 \\ 0.5250 & -0.0411 & -0.3276 \\ 0.5159 & -0.3262 & 0.3111 \end{bmatrix}$$

which can be used to compute the expected (I1, I2, I3) for a given set (I, Q, U). By inverting the appropriate matrix, the Stokes parameters (I, Q, U) can be computed from a set of observations (I1, I2, I3).

The errors on the Stokes parameters are determined by straightforward propagation of errors,

$$\sigma S_i^2 = \sigma_{I_1*}^2\left(\frac{\partial S_i}{\partial I_1*}\right)^2 + \sigma_{I_2*}^2\left(\frac{\partial S_i}{\partial I_2*}\right)^2 + \sigma_{I_3*}^2\left(\frac{\partial S_i}{\partial I_3*}\right)^2$$

where S_k represents an incoming Stokes vector, and I_k* defines the set of three observed intensities. Therefore,

$$\sigma_I^2 = a_{11}^2 \cdot \sigma_{I_1*}^2 + a_{12}^2 \cdot \sigma_{I_2*}^2 + a_{13}^2 \cdot \sigma_{I_3*}^2$$
$$\sigma_Q^2 = a_{21}^2 \cdot \sigma_{I_1*}^2 + a_{22}^2 \cdot \sigma_{I_2*}^2 + a_{23}^2 \cdot \sigma_{I_3*}^2$$
$$\sigma_U^2 = a_{31}^2 \cdot \sigma_{I_1*}^2 + a_{32}^2 \cdot \sigma_{I_2*}^2 + a_{33}^2 \cdot \sigma_{I_3*}^2$$

where a_{ij} represents the elements of the inverted transmission coefficients matrix.

[10] Further detailed information on calibration methodologies and transformations between different polarizers, see Mazzuca, Sparks, Axon, NICMOS ISR 98-017, "Methodologies to Calibrating NICMOS Polarimetry Characteristics, 1998.

The Stokes parameters can then be combined to yield the polarized intensity,

$$I_p = \sqrt{(Q^2 + U^2)}$$

as well as the degree of polarization P and the position angle of polarization θ_p, where

$$P = \frac{I_p}{I}, \quad \theta_p = \frac{1}{2}\operatorname{atan}\frac{U}{Q}$$

Because the polarizers are non-ideal, I, Q, and U are correlated when calculating P and θ_p. Therefore, covariances must be taken into account when calculating the errors.

For the degree of polarization, the covariance is

$$\sigma P^2 = \sigma_I^2\left(\frac{\partial P}{\partial I}\right)^2 + \sigma_Q^2\left(\frac{\partial P}{\partial Q}\right)^2 + \sigma_U^2\left(\frac{\partial P}{\partial U}\right)^2 +$$
$$2\sigma_{IQ}^2\left(\frac{\partial P}{\partial I}\right)\left(\frac{\partial P}{\partial Q}\right) + 2\sigma_{IU}^2\left(\frac{\partial P}{\partial I}\right)\left(\frac{\partial P}{\partial U}\right) + 2\sigma_{QU}^2\left(\frac{\partial P}{\partial Q}\right)\left(\frac{\partial P}{\partial U}\right)$$

where

$$\frac{\partial P}{\partial I} = \frac{\sqrt{Q^2 + U^2}}{I^2}, \qquad \frac{\partial P}{\partial Q} = \frac{Q}{I\sqrt{Q^2 + U^2}}, \text{ and } \quad \frac{\partial P}{\partial U} = \frac{U}{I\sqrt{Q^2 + U^2}}$$

The covariance for the position angle is

$$\sigma_{\phi_P}{}^2 = \sigma_Q^2\left(\frac{\partial \phi_P}{\partial Q}\right)^2 + \sigma_U^2\left(\frac{\partial \phi_P}{\partial U}\right)^2 + 2\sigma_{QU}^2\left(\frac{\partial \phi_P}{\partial Q}\right)\left(\frac{\partial \phi_P}{\partial U}\right)$$

where

$$\frac{\partial \phi_P}{\partial Q} = \frac{U}{2\left(\frac{U^2}{Q^2} + 1\right)Q^2}, \qquad \frac{\partial \phi_P}{\partial U} = \frac{1}{2\left(\frac{U^2}{Q^2} + 1\right)Q}$$

5.7.3 A Useful Script for Polarization Analysis

An interactive IDL program to derive relevant parameters from NICMOS polarization images has been developed. The IDL program reads three images taken with three polarizers from NIC1 or NIC2 produces five images as output. The output images are

- *q.fit* and *u.fit*, two images representing the Stokes parameters,
- *i.fit*, the total intensity,
- *p.fit*, the degree of polarization, and
- *theta.fit*, the polarization angle.

Polarization vectors or contour maps can be superimposed over the intensity image. The program is available from the STScI NICMOS web site under software tools, and is described in more detail in Mazzuca & Hines, NICMOS ISR 99-004, “User’s Guide to Polarimetric Imaging Tools.”

5.8 Grism Data Reduction

The NICMOS camera 3 grisms permit multi-object, slitless, low resolution spectroscopy. The reduction and analysis of NICMOS grism data benefit from decisions made by the user and from careful, interactive examination, and are therefore discussed here rather than in the chapter on pipeline calibration.

IDL software to extract spectra from NICMOS grism images has been developed at Space Telescope European Coordinating Facility (ST-ECF). Two programs are available: **NICMOSlook**, which is interactive, and **calnicc**, which is non-interactive. The interactive program, **NICMOSlook**, provides a number of tools called from an IDL GUI widget. This program is recommended for most extraction because of its versatility and interactive features. The automatic version, **calnicc**, is recommended if a quick-look extraction of a large number of spectra from different images is desired to identify interesting objects. Both software packages, together with user documentation, can be obtained from the ST-ECF NICMOS web site. Here we offer only a brief description of grism analysis methodology, and refer the user to the documentation provided with the software for details. The ST-ECF also maintains grism calibration reference files which are included with the software distributions.

Grism data are treated as regular images by the extraction software (see section 5.8.1) with the exception that they are not flatfielded by **calnica**. The flatfield correction for a given pixel depends both on the pixel location *(x,y)* and on the wavelength of the light which is dispersed onto that pixel. The latter is not known until the location of the dispersed source has been specified, and therefore the flatfielding cannot be done in advance. It is omitted during 2-dimensional data processing through **calnica**. Later, after 1-dimensional spectra have been extracted from the grism images, a wavelength dependent correction for QE variations is applied (see section 5.8.2).

The wavelength calibration for the extraction of spectra requires a direct image corresponding to each grism image. If individual exposures for the grism images are co-added before extraction, a similar co-addition should also be performed for the direct images in order to maintain the relative position of objects in the direct image with the corresponding spectrum in the grism image. If grism and direct images are processed separately by **calnicb**, this relative registration is, in general, not maintained. An IRAF package **nictools** which helps to co-add direct and grism images is available from the ST-ECF nictools web site. The package includes tasks to create association files (**makeasn**), find relative shifts from direct images and apply them to grism images (**alignasn**), and combine associated images with those fits applied (**asncombine**). This package can be used to prepare images for the extraction of the spectra.

In general, combining dithered grism images before extracting spectra is probably not a good idea whenever it can be avoided. As noted above, every pixel on the array has a different spectral response. Combining dithered grism images before extraction will combined data from different pixels, making it difficult or impossible to reliably flux calibrate the resulting spectrum. In general, we recommend that individual spectral and direct image pairs be reduced and the resulting spectra combined, when the individual spectra are strong enough.

5.8.1 Extraction Software

Detailed software manuals and descriptions of the extraction algorithms can be found at the above URLs. Only a brief summary is given below.

Input Files

The extraction software requires two types of input images, one for object finding, and one which contains the spectra to be extracted. Typically, the former is a direct image of the target field obtained with one of the NICMOS continuum filters, preferably at a wavelength within the range covered by the grism. However, the grism image itself can also be used for object finding, e.g. on the zeroth order spectra. The image which contains the spectra is assumed to be not flatfielded, which is the default in **calnica**.

The input files can be either the output of **calnica** (`*_cal.fits`) or the mosaiced images created with **calnicb** (`*_mos.fits`). The software also reads FITS images without the NICMOS specific extension, but some functionalities which depend on the error planes (`*_cal.fits[err]`) or data quality flags (`*_cal.fits[dq]`) will not be available.

Calibration data and configuration files for the extraction programs are in directory `calib`. The ASCII file `grismspec.dat` describes grism parameters, and `calnicc.setup` specifies configuration parameters. Parameters interactively changed and saved with **NICMOSlook** are written to those files. File names of other calibration data are specified in those two files.

Output Files

The basic output of the extraction are the spectra, which can be saved in several forms.

1. A FITS table containing the extracted spectra (`image_spc.fits`). This file contains the successfully extracted spectra; it consists of a primary header and a series of table extensions, each extension consisting of a header and the associated table. There is one table extension for each spectrum extracted. The primary header of the file contains the relevant information regarding the observation, namely a subset of the keywords in the primary headers of the input direct and grism images. The table extension header contains keywords relevant for the individual spectrum; the keywords describe: the content of the table (the list of columns), the nature and position of the object, and the characteristics of the spectrum (line positions and fluxes, continuum level, etc.). The associated table contains five columns: the wavelength vector, the flux vector, and three vectors of the statistical, deblending, and total errors from the extraction process.
2. Postscript files of the extracted spectra (`image_n.ps`). This file contains graphical representations of the extracted spectra. One postscript file for each spectrum is generated, where `n` is a sequential number starting with 0.

In addition, a number of miscellaneous output files can be created by the software, which include background image, finding charts, catalog files and object lists. In **NICMOSlook**, these files have to be explicitly requested, while in **calnicc** files are created depending on the configuration of the program.

5.8.2 Processing

Object Detection and Classification

Objects on the direct images may be found by interactively using the cursor. Automatic programs to find objects are also available. **NICMOSlook** uses **DAOFIND** for that purpose, while **calnicc** uses **Sextractor** to find and classify objects.

It may sometimes be useful to use the grism images to search for particular types of spectra "by eye". In this case, the spectral images can be flatfielded using ordinary, on-orbit grism flatfields, displayed, and examined visually for e.g. emission line objects, or very red spectra. Ordinarily, these on-orbit flatfields are *not* used as part of NICMOS grism data processing: instead, the flatfielding is done on the extracted spectra, as is described below. Once the interesting objects have been identified, their spectra should be extracted from non-flatfielded grism images.

Location of Spectra

The positions of the direct objects can be used to compute the location and orientation of the spectra. The positions of spectra relative to the position of the object on the direct image is parameterized in `grismspec.dat`. However, the orientation of spectra varies enough from observation to observation so that a "tracing" of spectra is necessary for accurate spectrum extraction. See the **NICMOSlook** manual for details.

Background Subtraction

After source identification, an estimate of the two-dimensional background level is derived and removed from each image.

The grism image is not flat-fielded and the QE variations across the NICMOS detectors are strong, implying that a significant structure is present in an image of blank sky. Several options to subtract this background are provided. They include interpolation over the region of the spectra, or subtracting scaled versions of background images. The extraction software determines the regions of interpolation excluding positions occupied by other spectra in the image.

The on-orbit grism flatfields, which are normally *not* used for grism data processing (see ***Flatfielding***, below), can provide one convenient way to artificially remove the background from spectral images before extracting the spectra. This is a particularly useful procedure if your field is crowded and other sources might contaminate the regions on either side of the spectra, used to determine the background. The grism flats can be significantly different from flats through regular filters, and do a better job of removing the background once they are inverted and scaled to match the background in the spectral image.

Extraction of Spectra

Flux and Error Bars

Once objects on the images have been detected, their spectra can be extracted. The flux is then given by:

$$F_j = \sum_i w_{ji} g_{ji}$$

where the sum over the flux g_{ji} of all pixels at wavelength λ is performed with weights w_{ji}.

Several options for the weights can be used to achieve optimum S/N. Constant weights lead to an optimum extraction for high S/N spectra, while for background limited objects, weights can be derived from the profile of the spectrum to be extracted. The profile can either be determined directly from the spectrum, or predicted from the direct image for very low S/N spectra under the assumption that the shape of the object is independent of wavelength. First, the size and orientation of the object is computed from the direct image using the moments of the image. The weights are then created by summing up all the pixel values in a given column (fixed wavelength) of the grism image that fall within the ellipse defined by the size and orientation of the object in the direct image.

Since NICMOS grism images are undersampled, spectra of point sources and sources up to the size of a few pixels are best extracted using constant weights even for low S/N spectra.

The error estimate e_{ji} for each pixel is taken from the `ERR` array of the input grism image. The error estimate e_j for each wavelength is then the weighted quadratic sum over the errors of all pixels at constant wavelength.

Wavelength Calibration

The dispersion relation and the deviation of the spectra have been determined from wavelength calibration observations, and are parametrized as:

$$\lambda = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

where x is the deflection in pixels relative to the position of the object in the direct image and λ is the corresponding wavelength. The coefficients are contained in the reference file `grismspec.dat`. The dispersion relation is given by:

$$\Delta y = b_0 + b_1 r + b_2 r^2 + b_3 r^3$$

where r is the distance of a pixel (x, y) from the object of coordinates (x_o, y_o) and Δy is the deviation in pixels of the spectrum from a horizontal line. The alignment of the spectrum is taken into account by rotating the grism image around the object position (x_o, $y_{o)}$ prior to the extraction. The distortions in the spectra are taken into account by introducing a corresponding distortion in the weights used for the extraction.

Flatfielding of Spectra

After the spectra are extracted, the fluxes are corrected for pixel-to-pixel variations in the quantum efficiency of the detector (i.e., flatfielded). The

QE variations depend both on the wavelength and on the position of the object on the detector. Because of this wavelength dependence, the flatfielding cannot be performed before the spectra are extracted and wavelength calibrated. The corrected flux $f_c(\lambda)$ is computed as follows

$$f_c(x,y) = f(x,y)q(x,y,\lambda)$$

where $q(x,y,\lambda)$ are interpolated flatfields. For wavelengths where narrow band flatfields are available, they are used. For other wavelengths, the correction factors are derived through interpolation from a set of monochromatic flatfield images (see Storrs et al. 1999, NICMOS ISR-99-002). The list of flatfields to be used is specified in `calnicc.setup`, and users can provide their own preferred flatfields. The default list is shown in table 5.10.

Table 5.10: Default Flatfields For Spectra

Flatfield File	λχ (μ)	FWHM (μ)	Filter
i191346kn_flt.fits	1.07990	0.0096000	F108N
i191346mn_flt.fits	1.12830	0.0110000	F113N
i191346pn_flt.fits	1.64600	0.0170000	F164N
i191346qn_flt.fits	1.65820	0.0164000	F166N
i191346sn_flt.fits	1.87380	0.0192000	F187N
i191346tn_flt.fits	1.90030	0.0174000	F190N
i1913470n_flt.fits	1.96390	0.0186000	F196N
i1913471n_flt.fits	1.99740	0.0206000	F200N
i1913472n_flt.fits	2.12130	0.0206000	F212N
i1913473n_flt.fits	2.14870	0.0200000	F215N
i1913475n_flt.fits	2.39770	0.1975000	F240M

Flux calibration and Correction for Pixel Response Function

Once the spectra are extracted, the count rates in ADU/second are converted to physical units using calibration data form photometric standards P330E and G191B2B.

Undersampling of NICMOS grism images in combination with significant variation of the QE across any given pixel imposes a wave-like pattern onto extracted spectra of point sources and small objects. Since spectra are not exactly aligned with the rows of the images, the exact sub-pixel position and orientation of the spectra determines the phase and period of those waves. A simple model can be used to correct this effect for point sources. Details are discussed in an article by W. Freudling in the May 1999 issue of the ST-ECF newsletter.

Deblending of Overlapping Spectra

Since the NICMOS grisms are slitless, overlaps among different spectra are likely to happen. The strategy of observing the same target at different telescope roll angles helps remove overlap in many instances. The extraction software includes an algorithm designed to remove or minimize contamination of one spectrum from another.

The deblending algorithm is described in detail in the **NICMOSlook** software manual. The basic requirement for the algorithm to work is that, at each wavelength, different spatial portions of the spectrum to be deblended have different levels of contamination. The deblending algorithm relies on the assumption that the shape of the object is the same at all wavelengths. The deblending procedure also produces an error estimate which is reported in the output FITS table and indicated in the postscript file containing the spectrum.

Part III:

Appendixes

Part III:Appendixes

APPENDIX A:

IRAF Primer

In this appendix . . .

The Image Reduction and Analysis Facility (IRAF), developed by the National Optical Astronomy Observatories (NOAO), forms the basis of the Space Telescope Science Data Analysis System (STSDAS). IRAF contains numerous packages of programs, called *tasks*, that perform a wide range of functions from reading data tapes to producing plots and images. Most astronomers will already be familiar with IRAF, but we provide this tutorial for HST observers who are beginners with IRAF. It includes information on:

- How to set up IRAF the first time you use the software.
- How to start and stop an IRAF session.
- Basic concepts, such as loading packages, setting parameters, etc.
- How to use the on-line help facility.

Additional information on IRAF, in particular *A Beginner's Guide to Using IRAF* is available through the NOAO IRAF Home Page at: http://iraf.noao.edu

A.1 Initiating IRAF

This section explains:

- How to set up your IRAF working environment.
- How to start and logout of the IRAF program.

We assume that your site has IRAF and STSDAS installed. If not, you must obtain and install the software. See appendix section A.3 for details.

A.1.1 Setting Up IRAF

Before running IRAF for the first time you need to follow these three steps:

1. Create your IRAF root directory.
2. Move to that directory and set the necessary environment variables or system logicals and symbols.
3. Run **mkiraf** to create a `login.cl` file and a `uparm` subdirectory.

Users generally name their IRAF home directory `iraf` (also referred to as your IRAF *root* directory) and set it up in their account's root directory (i.e., the default directory that you are in when you log in to the system). The IRAF home directory doesn't need to be in your account's root directory, nor does it need to be called `iraf`, but you should *not* put it on a scratch disk that is periodically erased.

If you call your root IRAF directory `"iraf"`, you can set up IRAF as follows:

Under Unix:

```
% mkdir iraf
% cd iraf
% setenv iraf /usr/stsci/iraf/
% source $iraf/unix/hlib/irafuser.csh
% mkiraf
```

Can be placed in .login file

The directory name is site-dependent—check with your system staff

Under VMS:

```
$ CREATE/DIR [.IRAF]
$ SET DEFAULT [.IRAF]
$ IRAF
$ MKIRAF
```

Can be placed in LOGIN.COM file

The **mkiraf** command initializes IRAF by creating a `login.cl` file and a subdirectory called `uparm`. After typing the **mkiraf** command, you will see the following:

```
% mkiraf
-- creating a new uparm directory
Terminal types: gterm=ttysw+graphics,vt640...
Enter terminal type:
```

Enter the type of terminal or workstation you will most often use with IRAF. [1] Generic terminal types that will work for most users are:

- `vt100` for most terminals.
- `xtermjhs` for most workstations running under X-Windows.

`xgterm` for sites that have installed X11 IRAF and IRAF v2.10.3 BETA or later.

You can change your terminal type at any time by typing set term=new_type during an IRAF session. You can also change your default type by editing the appropriate line in your login.cl file.

After you enter your terminal type, you will see the following output before getting your regular prompt:

```
A new LOGIN.CL file has been created in the current ...
You may wish to review and edit this file to change ...
```

The `login.cl` file is the *startup file* used by the IRAF command language (CL). It is similar to the `LOGIN.COM` file used by VMS or the `.login` file used by Unix. Whenever IRAF starts, it looks at the `login.cl` file. You can edit this file to customize your IRAF environment. In fact, you should look at it to make sure that everything in it is correct. In particular, there is a line starting with `set home =` that tells IRAF where to find your IRAF home directory. You should verify that this statement does, in fact, point to your IRAF directory. If you will be working with standard IRAF format images you should also insert a line saying `set imdir = "HDR$"`. The `imdir` setting is ignored when working with GEIS format images.

The `uparm` directory will contain your own copies of IRAF task parameters. This directory allows you to customize your IRAF

[1] Users at STScI should consult the *STScI Site Guide for IRAF and STSDAS*.

environment by setting certain parameter values as defaults. Once you set up IRAF, you should rarely need to do it again, expect when updated version of IRAF are installed.

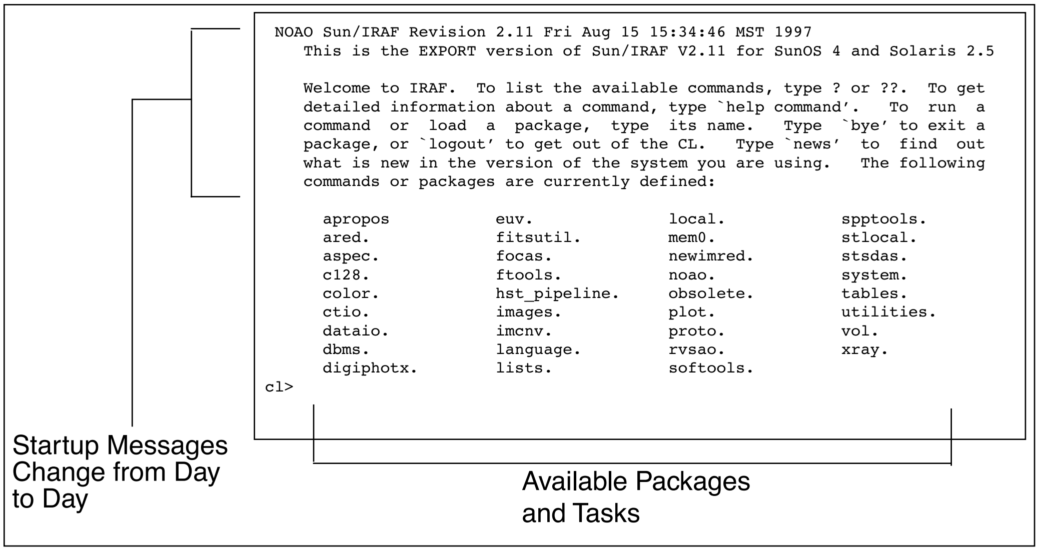
A.1.2 Starting and Stopping an IRAF Session

To start an IRAF session:

1. Move to your IRAF home directory.
2. Type `cl`.

IRAF starts by displaying several lines of introductory text and then puts a prompt at the bottom of the screen. Figure A.1 is a sample IRAF startup screen.

Figure A.1:IRAF Startup Screen



```
  NOAO Sun/IRAF Revision 2.11 Fri Aug 15 15:34:46 MST 1997
     This is the EXPORT version of Sun/IRAF V2.11 for SunOS 4 and Solaris 2.5

     Welcome to IRAF.  To list the available commands, type ? or ??.  To get
     detailed information about a command, type `help command'.   To  run  a
     command  or  load  a  package,  type  its name.   Type  `bye' to exit a
     package, or `logout' to get out of the CL.   Type `news'  to  find  out
     what is new in the version of the system you are using.   The following
     commands or packages are currently defined:

       apropos          euv.             local.           spptools.
       ared.            fitsutil.        mem0.            stlocal.
       aspec.           focas.           newimred.        stsdas.
       c128.            ftools.          noao.            system.
       color.           hst_pipeline.    obsolete.        tables.
       ctio.            images.          plot.            utilities.
       dataio.          imcnv.           proto.           vol.
       dbms.            language.        rvsao.           xray.
       digiphotx.       lists.           softools.
cl>
```

Startup Messages Change from Day to Day
Available Packages and Tasks

To quit an IRAF session:

1. Type `logout`.

A.2 IRAF Basics

This section describes basic IRAF techniques such as:

- Loading packages (below).
- Running tasks and commands.
- Getting online help.
- Viewing and setting parameters (see appendix section A.2.4).

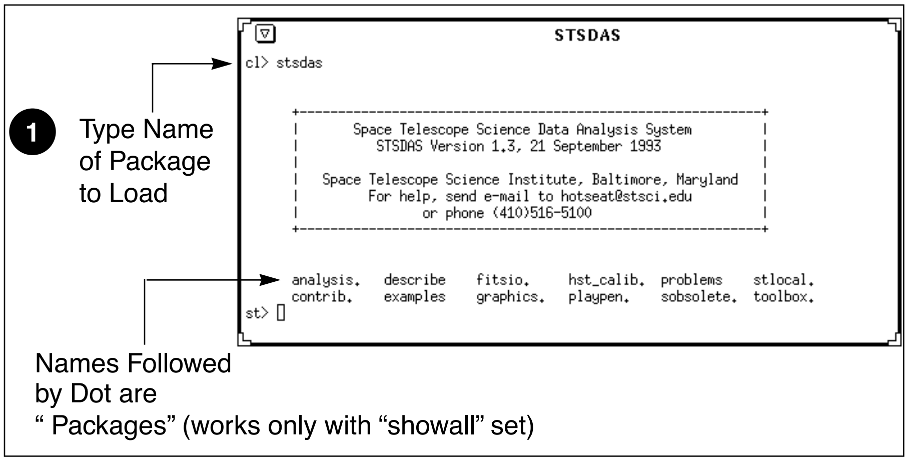
- Setting and using environment variables (see section A.2.5).
- File management
- Troubleshooting

A.2.1 Loading Packages

In IRAF jargon, an application is called a *task* and logically related tasks are grouped together in a *package*. Before you can use a task, you must load the package containing that task. To load a package, type the name of the package. The prompt will then change to the first two letters of the package name, and the screen will display the names of all the newly available tasks and subpackages. Even though the prompt has changed, previously loaded packages remain loaded, and all their tasks remain available.

Note that the standard way to specify a path through the IRAF package hierarchy to a task in a particular subpackage is to separate the package names with periods (e.g., **stsdas.hst_calib.foc.focgeom.newgeom**).

Figure A.2:Loading Packages



STSDAS
1
Type Name of Package to Load
Names Followed by Dot are
“ Packages” (works only with “showall” set)

```
cl> stsdas


      +------------------------------------------------------------+
      |      Space Telescope Science Data Analysis System          |
      |         STSDAS Version 1.3, 21 September 1993              |
      |                                                            |
      |   Space Telescope Science Institute, Baltimore, Maryland   |
      |         For help, send e-mail to hotseat@stsci.edu         |
      |               or phone (410)516-5100                       |
      +------------------------------------------------------------+


      analysis.   describe    fitsio.     hst_calib.  problems    stlocal.
      contrib.    examples    graphics.   playpen.    sobsolete.  toolbox.
st> 
```

Some helpful commands for managing packages are:

- `?` - Lists tasks in the most recently-loaded package.
- `??` - Lists all tasks loaded, regardless of package.
- `package` - Lists names of all loaded packages.
- `bye` - Exits the current package.

A.2.2 Running Tasks

This section explains how to run tasks, background tasks, and system-level commands, and how to use piping and redirection.

Running a Task

The simplest way to run a task is to type its name or any unambiguous abbreviation of it. The task will then prompt you for the values of any required *parameters*, such as the names of input files. Alternatively, you can specify the values for the required *parameters* on the command line when you run the task. For example, if you want the task `imheader` to print header information on the file `myfile.hhh`, you can type

```
st> imhead myfile.hhh
```

IRAF does not require you to type the complete command name—only enough of it to make it unique. For example, dir is sufficient for directory.

Escaping System-Level Commands

To run an operating system-level command (i.e., Unix or VMS commands) from within the IRAF CL, precede the command with an exclamation point (!). This procedure is called *escaping* the command. For example:

```
st> !system_command
```

Piping and Redirection

You can run tasks in sequence if you desire, with the output of one task being used as the input for another. This procedure, called *piping*, and is done by separating commands with a vertical bar (|), using the following syntax:

```
st> task1 filename | task2
```

For example, if a particular task prints a large volume of textual output to the screen, you will often want to pipe it to `page`, which allows you to read the output one page at a time:

```
st> task1 filename | page
```

You can also redirect output from any task or command to a file by using the greater-than symbol (>) as follows:

```
st> command > outputfile
```

Background Tasks

To run a task as a background job, freeing your workstation window for other work, add an ampersand (&) to the end of the command line, like this:

```
st> taskname &
```

A.2.3 Getting Help

This section describes:

- How to use IRAF’s on-line help facility.
- How to find a task that does what you want (see “Finding Tasks” on page A-8).

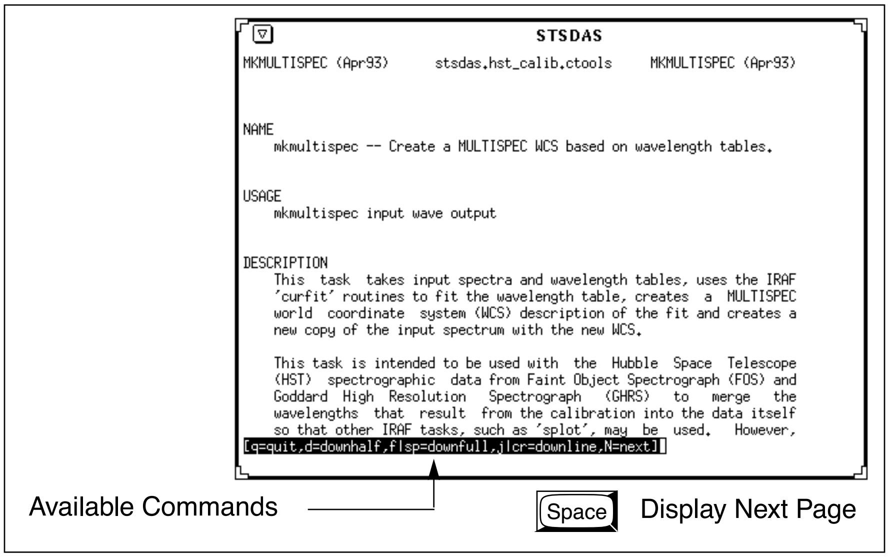
On-Line Help

You can get on-line help with any IRAF task or package by using the **help** command,[2] which takes as an argument the task or package name about which you want help. Wildcards are supported. For example, to display the on-line help for the STSDAS **mkmultispec** task, you would type:

```
fi> help mkmultispec
```

[2] There is an optional *paging* front-end for help called **phelp**. For more information, type `help phelp` from within IRAF.

Figure A.3:Displaying On-line Help



```
                              STSDAS
MKMULTISPEC (Apr93)      stsdas.hst_calib.ctools     MKMULTISPEC (Apr93)



NAME
    mkmultispec -- Create a MULTISPEC WCS based on wavelength tables.


USAGE
    mkmultispec input wave output


DESCRIPTION
    This  task  takes input spectra and wavelength tables, uses the IRAF
    'curfit' routines to fit the wavelength table, creates  a  MULTISPEC
    world  coordinate  system (WCS) description of the fit and creates a
    new copy of the input spectrum with the new WCS.

    This task is intended to be used with  the  Hubble  Space  Telescope
    (HST)  spectrographic  data from Faint Object Spectrograph (FOS) and
    Goddard  High  Resolution   Spectrograph   (GHRS)   to   merge   the
    wavelengths  that  result  from the calibration into the data itself
    so that other IRAF tasks, such as 'splot', may  be  used.  However,
[q=quit,d=downhalf,f|sp=downfull,j|cr=downline,N=next]
```

Two STSDAS tasks that display only certain sections of the help file are also available:

- **examples** - Displays only the examples for a task.
- **describe** - Displays only the description of the task.

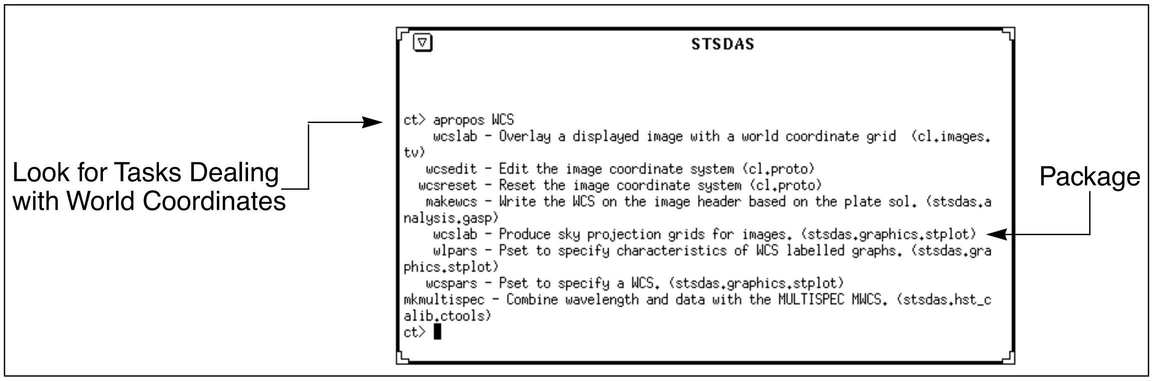
Typing `help` *package* will produce one-line descriptions of each task in the package.

Finding Tasks

There are several ways to find a task that does what you need:

- Use `help` *package* to search through the IRAF/STSDAS package structure.
- Use the **apropos** task as shown in figure A.4 to search the online help database. This task looks through a list of IRAF and STSDAS package menus to find tasks that match a specified keyword. Note that the name of the package containing the task is shown in parentheses.
- Ask more experienced user, who can usually point you in the right direction.

Figure A.4:The **apropos** task Using apropos



Look for Tasks Dealing with World Coordinates

Package

```
ct> apropos WCS
    wcslab - Overlay a displayed image with a world coordinate grid  (cl.images.
tv)
   wcsedit - Edit the image coordinate system (cl.proto)
  wcsreset - Reset the image coordinate system (cl.proto)
   makewcs - Write the WCS on the image header based on the plate sol. (stsdas.a
nalysis.gasp)
    wcslab - Produce sky projection grids for images. (stsdas.graphics.stplot)
    wlpars - Pset to specify characteristics of WCS labelled graphs. (stsdas.gra
phics.stplot)
   wcspars - Pset to specify a WCS. (stsdas.graphics.stplot)
mkmultispec - Combine wavelength and data with the MULTISPEC MWCS. (stsdas.hst_c
alib.ctools)
ct> █
```

A.2.4 Setting Parameters

Parameters specify the input information for IRAF tasks. They can be the names of input or output files, particular pixel numbers, keyword settings, or many other types of information that control the behavior of the task.

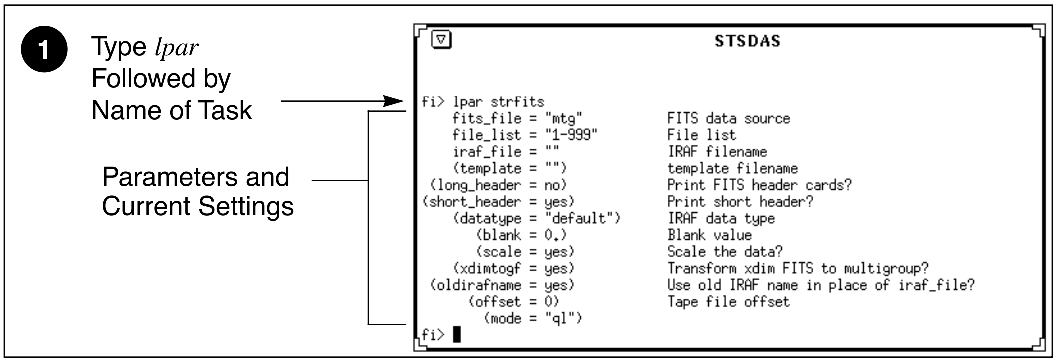
The two most useful commands for handling parameters are:

- **lparam** to display the current parameter settings (often abbreviated **lpar**).
- **eparam** to edit parameters (often abbreviated **epar**).

Viewing Parameters with lparam

The **lpar** command lists the current parameter settings for a given task (figure A.5).

Figure A.5:Displaying Parameter Settings with lpar



1 Type *lpar* Followed by Name of Task

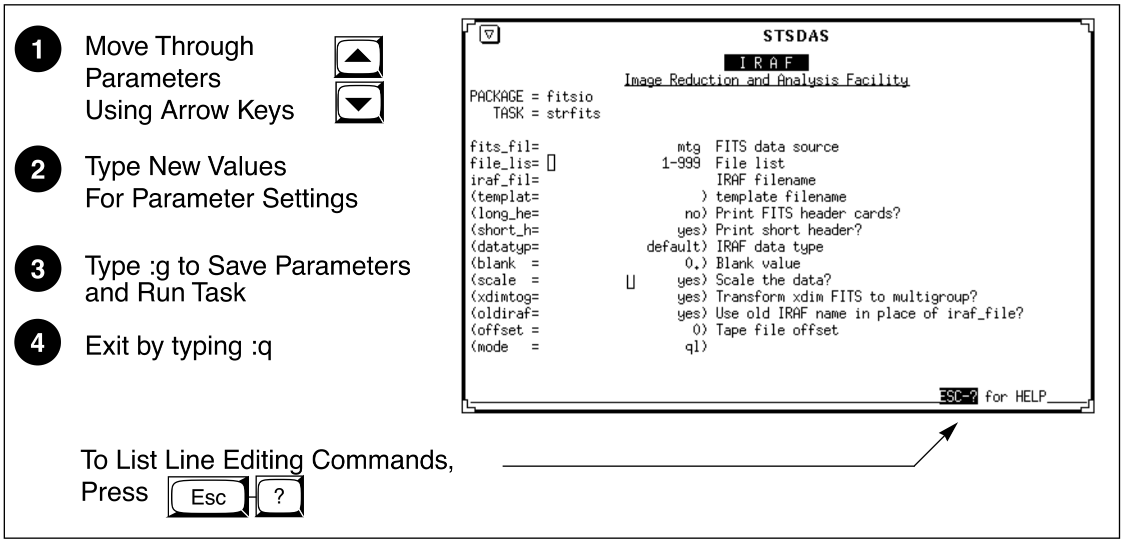
Parameters and Current Settings

```
fi> lpar strfits
    fits_file = "mtg"          FITS data source
    file_list = "1-999"        File list
    iraf_file = ""             IRAF filename
    (template = "")            template filename
 (long_header = no)            Print FITS header cards?
(short_header = yes)           Print short header?
    (datatype = "default")     IRAF data type
       (blank = 0.)            Blank value
       (scale = yes)           Scale the data?
    (xdimtogf = yes)           Transform xdim FITS to multigroup?
 (oldirafname = yes)           Use old IRAF name in place of iraf_file?
      (offset = 0)             Tape file offset
        (mode = "ql")
fi> █
```

Setting parameters with eparam

The **epar** command is an interactive parameter set editor. It displays all of the parameters and their current settings on the screen. You can move around the screen using the arrow keys (also called *cursor* keys) and type new settings for any parameters you wish to change. Figure A.6 shows a sample of the **epar** editor at work (invoked by typing `epar strfits`).

Figure A.6:Editing Parameters with epar



1 Move Through Parameters Using Arrow Keys ▲ ▼

2 Type New Values For Parameter Settings

3 Type :g to Save Parameters and Run Task

4 Exit by typing :q

To List Line Editing Commands, Press Esc ?

```
                                STSDAS
                                I R A F
                    Image Reduction and Analysis Facility
PACKAGE = fitsio
   TASK = strfits

fits_fil=                  mtg  FITS data source
file_lis=                1-999  File list
iraf_fil=                       IRAF filename
(templat=                     ) template filename
(long_he=                   no) Print FITS header cards?
(short_h=                  yes) Print short header?
(datatyp=              default) IRAF data type
(blank  =                   0.) Blank value
(scale  =                  yes) Scale the data?
(xdimtog=                  yes) Transform xdim FITS to multigroup?
(oldiraf=                  yes) Use old IRAF name in place of iraf_file?
(offset =                    0) Tape file offset
(mode   =                   ql)

                                                          ESC-? for HELP
```

Parameter Data Types—What to Specify

Parameters are either *required* or *hidden*, and each parameter expects information of a certain *type*. Usually, the first parameter is required, and very often it expects a file name. Parameters are described in the online help for each task [include reference to help]. Hidden parameters, shown in parentheses in the online help and the **lpar** and **epar** listings, need not be specified at each execution because their default values frequently suffice.

▽ ***Wise IRAF users will check the values of hidden parameters, as they often govern important aspects of a task's behavior.***

If you specify the wrong type of information for a parameter, **epar** will usually display an error message saying something like “Parameter Value is Out of Range.” The message is displayed when you move to another parameter or if you press Return. Table A.1 lists the different parameter types.

Table A.1: Parameter Data Types

Type	Description
`File Name`	Full name of the file. Wild card characters (* and ?) are often allowed. Some tasks allow you to use special features when specifying file names, including “@” lists, IRAF networking syntax, and image section or group syntax. (See “File Management” below).
`Integer`	Whole number. Often the task will specify minimum or maximum values (see the help pages).
`Real`	Floating point numbers, can be expressed in exponential notation. Often will have minimum and maximum values.
`Boolean`	Logical “yes” or “no” values.
`String`	Any characters. Sometimes file names are specified as string.
`Pset`	Parameter set.

Restoring Parameter Default Values

Occasionally, IRAF (or you) will get confused by your parameter values. To alleviate this confusion, you can restore the default parameters with the **unlearn** command. You can use **unlearn** on either a task or on an entire package.

The unlearn command generally will restore the parameters to reasonable values, a big help if you are no longer sure which parameter values you have changed in a complicated task.

A.2.5 Setting Environment Variables

IRAF uses *environment variables* to define which devices are used for certain operations. For example, your terminal type, default printer, and the disk and directory used for storing images are all defined through environment variables. Environment variables are set using the **set** command and are displayed using the **show** command. Table A.2 lists some of the environment variables that you might want to customize.

Table A.2: Environment Variables

Variable	Description	Example of Setting
`printer`	Default printer for text	`set printer = lp2`
`terminal`	Terminal type	`set term = xterm`
`stdplot`	Default printer for all graphics output	`set stdplot = ps2`
`stdimage`	Default terminal display setting for image output (most users will want this set to either imt512 or imt800)	`set stdimage = imt800`
`stdgraph`	Default graphics device	`set stdgraph = xterm`
`clobber`	Allow or prevent overwriting of files	`set clobber = yes`
`imtype`	Default image type for output images. “imh” is original IRAF format, “hhh” is STSDAS GEIS format.	`set imtype = “hhh”`

If you are working with GEIS files, you should set imtype to “hhh”. If you are working with STIS and NICMOS data in FITS files, you can set imtype to “fits”

You can set your environment variables automatically each time you login to IRAF by adding the appropriate commands to your `login.cl` file. Use your favorite text editor to specify each variable on its own line. The **show** command with no arguments prints the names and current values of all environment variables.

A.2.6 File Management

This section describes:

- File formats commonly used with STSDAS and IRAF.
- Specification of file names.
- Navigation through directories.

File Formats

IRAF recognizes a number of different file structures. Among them are the standard HST file formats known as GEIS and FITS (see chapter 2 of the HST Introduction), both of which differ from the original IRAF format (OIF). GEIS is closer to OIF, in that two files are *always* used together as a pair:

- A *header file*, which consists of descriptive information. IRAF header files are identified by the suffix `.imh`. GEIS header files are in ASCII text format and are identified by the suffix `.hhh` or another suffix ending in “h”, such as `.c0h` or `.q1h`.
- A *binary data file*,[3] consisting of pixel information. IRAF data file names end with a `.pix` suffix. STSDAS data files end with an suffix of `.hhd` or another suffix that ends with “d”, such as `.c0d` or `.q0d`.

STSDAS always expects both component files of a GEIS image to be kept together in the same directory. A single FITS file contains both the header information and the data.

When working with IRAF or STSDAS images, you need only specify the header file name—the tasks will automatically use the binary data file when necessary.

File Specification

Most tasks in IRAF and STSDAS operate on files and expect you to specify a file name for one or more parameters. Several types of special syntax can be used with certain tasks when specifying file names. These syntax features include:

- **Wild card characters**, often called *templates*, which are used to specify multiple files using pattern matching techniques. The wild cards are:
  - * Matches any number of characters, e.g.: `z*.c0h`
  - ? Matches any single character, e.g.: `z01x23x.c?h`

[3] The binary data file format is host-dependent and may require translation before it can be moved to a computer using a different architecture.

When using wildcards with image-processing tasks, be sure to exclude the binary pixel files by ending your file name specification with an "h", for example: y*.??h

- **List files**, often called *@-files*, which are ASCII file that contain lists of file names, one per line. If your task supports the list file feature, you would type the name of your list file, preceded by the "@" character. For example: `@files.txt`
- **Image section** specification. Tasks that work with image data will often let you specify that you want to work on only a small area of the image rather than the entire image. To extract a particular image section, specify each axis range in square brackets, for example: `image.hhh[10:200,20:200]`
- **IRAF networking** specification. IRAF is capable of reading and writing files to and from remote systems on a network. This feature is often used with tasks in the **fitsio** and **convfile** packages, or with image display tasks. The *STSDAS Users Guide* and the online help (type `help networking`) describe how to enable this feature. To specify that you want to use the IRAF networking feature, type the remote host name followed by an exclamation point (!), followed by the file or device name. For example: `ra!mta`.

Directory Navigation

To navigate through directories, you can use the following commands:

- **path** or **pwd** - Lists the current working directory.
- **cd** *directory* - Move to the named directory.

A.2.7 Troubleshooting

There are a couple of easy things you can do to make sure that you don't have a simple memory or parameter conflict—common causes of problems.

- Look at the parameter settings and make sure that you have specified reasonable values for every parameter.

- When you run an IRAF task for the first time in a session, IRAF stores the executable file in its *process cache*. If IRAF appears not to be running your tasks properly, you may need to use the **flprcache** command to clear the process cache. To do this type: `flpr` Sometimes you will need to execute this command twice in succession.
- Occasionally, you may need to logout of the CL, restart IRAF, and try your command again.

If you still have a problem, contact the STScI Help Desk at help@stsci.edu

A.3 Getting IRAF and STSDAS

Both IRAF and STSDAS are provided free of charge to the astronomical community. You must have IRAF to run STSDAS. Detailed information about installing and retrieving STSDAS is found in the *STSDAS Site Manager's Installation Guide and Reference*. If you have any problems getting and installing STSDAS, TABLES, or any other packages or data described in this handbook, please contact the Help Desk by sending e-mail to: help@stsci.edu.

A complete description of how to install the **synphot** data files is provided in section A.3.2.

A.3.1 Retrieving the IRAF and STSDAS Software

There are three ways to get the software:

- Use the World Wide Web.
- Use anonymous FTP.
- Request a tape.

World Wide Web

The STSDAS World Wide Web page:

http://stsdas.stsci.edu/STSDAS.html

provides links and instructions for downloading the appropriate files to your local system or to display the software directory, from which you can select the series of smaller files.

Anonymous FTP

- **IRAF**: `iraf.noao.edu` (140.252.1.1)
- **STSDAS**: `ftp.stsci.edu` (130.167.1.2)

There are two points to remember when using FTP to retrieve STSDAS:

- You must retrieve and install the TABLES package before STSDAS.
- You should retrieve the `README` file from the directory `/software/ stsdas/v2.0` and read it to find out which files you should retrieve.

You must have IRAF installed on your system to install TABLES and STSDAS. When you retrieve STSDAS, you must also retrieve the TABLES package, and TABLES must be installed first.

Instructions for installing STSDAS are available in the `doc` subdirectory of the directory where you find STSDAS. The complete instructions for installing STSDAS, TABLES, and all of the supporting software and reference files (including instrument reference files and the **synphot** dataset) are found in the *STSDAS Site Manager's Installation Guide and Reference*.

Registration

The software can also be registered and requested using on-line forms available through World Wide Web at the following URL:

`http://stsdas.stsci.edu/RegistForm.html`

When you request the STSDAS software, you can also ask for the appropriate version of IRAF, which will be requested for you— simply check the appropriate box on the form under "Do You Already Have IRAF Installed?" If you prefer to request the IRAF software independent of STSDAS, you can do so by sending e-mail to: `iraf@iraf.noao.edu`

A.3.2 Getting the Synphot Database

This manual sometimes refers to the **synphot** dataset, which must be available in order to run tasks in the STSDAS **synphot** package. These data files are not included with the STSDAS software and must be retrieved independently. To do this, you need to retrieve a series of compressed tar files from the STScI FTP site (ftp.stsci.edu) in the directory `software/stsdas/refdata/synphot`. After uncompressing and extracting the tar files (see below), you need to unpack the FITS files as described below.

The synthetic photometry data are read in similar way as the instrument datasets, using the script `unpack.cl` provided in the top directory. This script is run within IRAF to convert data from FITS format into the format used by the **synphot** task. This script assumes you have the logical crrefer set up in your extern.pkg file (which is in the directory $iraf/unix/hlib (Unix) or $iraf/vms/hlib (VMS)) or have it set up in your session. You do this by placing the command below in extern.pkg or by typing it on the command line:

```
set crrefer = "/node/partition/stdata/synphot/"
```

Figure A.7 shows how to convert the files.

Figure A.7:Unpacking Synthetic Photometry Files

```
% cl
cl> cd /node/partition/stdata/synphot
cl> set crrefer = "/node/partition/stsdata/synphot/"
cl> task $unpack = unpack.cl
cl> tables
ta> fitsio
fi> unpack
```

Just in case...

The “$” is used because the task has no parameter file

Note that all three synphot files must be unloaded for the script to complete successfully.

A.3.3 Extracting the synphot Unix Tar Files

If you retrieved the **synphot** database as compressed tar files, you will need to copy them to an appropriate subdirectory and then expand and unpack the files. The tar and compress utilities that do this are are commonly available on most Unix systems, but are not standard in the VMS environment. The examples shown below reflect Unix usage. If you are on a VMS system, you should consult with your systems support staff regarding the availability and usage of these commands. To process the files on a Unix system:

1. Get the compressed tar file that you want, as described in previous sections.
2. Make an appropriate subdirectory using the mkdir command.
3. Pipe the compressed tar file through the uncompress and tar files to expand and unpack the file.

The following example shows how to do this. The example assumes that you are putting the files in a subdirectory under /usr/iraf/stdata (note that the name of your file here is assumed to be XXX.tar.Z).

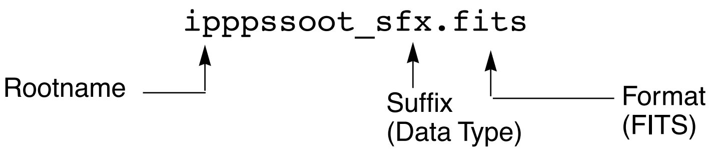
```
% pwd
/usr/iraf/stdata
% mkdir XXX
% mv XXX.tar.Z XXX/
% cd XXX
% cat XXX.tar.Z | uncompress | tar -xf -
```

APPENDIX B:

HST File Names

In this appendix. . .

This appendix describes the syntax of HST data file names, which encode a large amount of information about the files themselves. Datasets retrieved from the Archive as described in consist of multiple files in FITS format, each with a name that looks like this:



- ***Rootname***: The first part of the file name (`ipppssoot`) is the *rootname* of the dataset to which the file belongs. All files belonging to a given dataset share the same rootname.
- ***Suffix***: The three-character second part of the name (`sfx`) is called the *suffix*, and it indicates the type of data the file contains.
- ***Format***: The identifier `.fits` indicates that this file is in FITS format.

For example, an FOC data file named `x3l80101t_d0f.fits` is a FITS file belong to the dataset with rootname `x3l80101t`, and its suffix `d0f` indicates that it contains raw science data.

In order to use IRAF/STSDAS tasks to work with data from instruments other than NICMOS and STIS, you will want to convert these FITS files into GEIS format. See section 2.2 in the HST Introduction for instructions on how to convert FITS files to GEIS files using **strfits**. Like FITS files, the names of GEIS files also derive from a file's rootname and suffix, and they look like this:

```
ipppssoot.sfx
```

Generally the suffixes of GEIS files end either in "`d`", indicating a binary data file, or "`h`", indicating an ASCII header file. The two GEIS files `x3l80101t_d0h` and `x3l80101t_d0d` together contain the same information as the single FITS file `x3l80101t_d0f.fits`.

The identifier referred to here as a "suffix" has often been called an "extension" in the past. However, the individual pieces of FITS files are also known as "extensions" (see section 2.2.1 in the HST Introduction). For clarity, this handbook will use the term "extension" when refering to a component of a FITS file and the term "suffix" when referring to the three character identifier in a filename.

B.1 Rootnames

Rootnames of HST data files follow the naming convention defined in table B.1, which expands on the previous convention as follows: an initial "N" indicates a NICMOS exposure, an intial "O" indicates a STIS exposure, and the rootnames of files containing association products (see below) end in a number (0-8).

Table B.1: IPPPSSOOT Root File Names

Character	Meaning
I	Instrument used, will be one of: ***E*** - Engineering data ***F*** - Fine Guidance Sensors ***N*** - Near Infrared Camera and Multi-Object Spectrograph ***O*** - Space Telescope Imaging Spectrograph ***S*** - Engineering subset data ***T*** - Guide star position data ***U*** - Wide Field/Planetary Camera-2 ***V*** - High Speed Photometer ***W*** - Wide Field/Planetary Camera ***X*** - Faint Object Camera ***Y*** - Faint Object Spectrograph ***Z*** -Goddard High Resolution Spectrograph
PPP	Program ID; can be any combination of letters or numbers (46,656 combinations possible). There is a unique association between program ID and proposal ID.
SS	Observation set ID; any combination of letters or numbers (1,296 possible combinations).
OO	Observation ID; any combination of letters or numbers (1,296 possible combinations).
T	Source of transmission or association product number ***M*** - Merged real time and tape recorded ***N*** - Retransmitted merged real time and tape recorded ***O*** - Retransmitted real time (letter 'O') ***P*** - Retransmitted tape recorded ***R*** - Real time (not recorded) ***T*** - Tape recorded ***0*** - Primary association product (number zero) ***1-8*** - NICMOS background association product

B.2 Suffixes of Files Common to all Instruments

The three-character suffix of a data file (e.g., d0h) identifies the type of data that a file contains. Because the meanings of these suffixes change from instrument to instrument, please refer to the appropriate instrument-specific Data Structures chapter for their definitions. Several types of file suffixes are, however, common to all instruments.

OMS Files

Observatory Monitoring System (OMS) files, having suffixes cm* or ji*, contain Observation Logs describing how the HST spacecraft behaved during a given observation. OMS headers, which you can read with the IRAF task **imheader** (see section 2.3.3 in the HST Introduction), are divided into groups of keywords that deal with particular topics such as SPACECRAFT DATA, BACKGROUND LIGHT, POINTING CONTROL DATA, and LINE OF SIGHT JITTER SUMMARY. The headers themselves provide short descriptions of each keyword.OMS tables and images record spacecraft pointing information as a function of time. For more information on OMS files, you can consult Appendix C or the STScI Observation Logs WWW pages at:

`http://www.stsci.edu/ftp/instrument_news/Observatory/obslog/OL_1.html`

PDQ Files

The suffix `pdq` denotes Post Observation Summary and Data Quality Comment files—*PDQ files*—which contain predicted as well as actual observation parameters extracted from the standard header and science headers. These files may also contain comments on any obvious features in the spectrum or image, as noted in the OPUS data assessment, or automatically extracted information about problems or oddities encountered during the observation or data processing. These comments may include correction to the keywords automatically placed in the OMS files.

OCX Files

The suffix `ocx` denotes Observer Comment Files—*OCX files*—which are produced by STScI personnel to document the results of real-time commanding or monitoring of the observation, along with keywords and comments. Prior to April 17, 1992, OCX files were not always archived separately and, in some cases, were prepended to the trailer file.

After early February 1995, OCX files were produced only when an observation was used to locate the target for an Interactive Target Acquisition. At this time, mission and spacecraft information were moved to the PDQ reports and the Observation Logs (OMS jitter image and jitter table).

Trailer Files

Trailer files (suffix `trl`) are FITS ASCII tables that log the processing of your data by the OPUS pipeline.

Note that trailer files are formatted with 132 columns

B.3 Associations

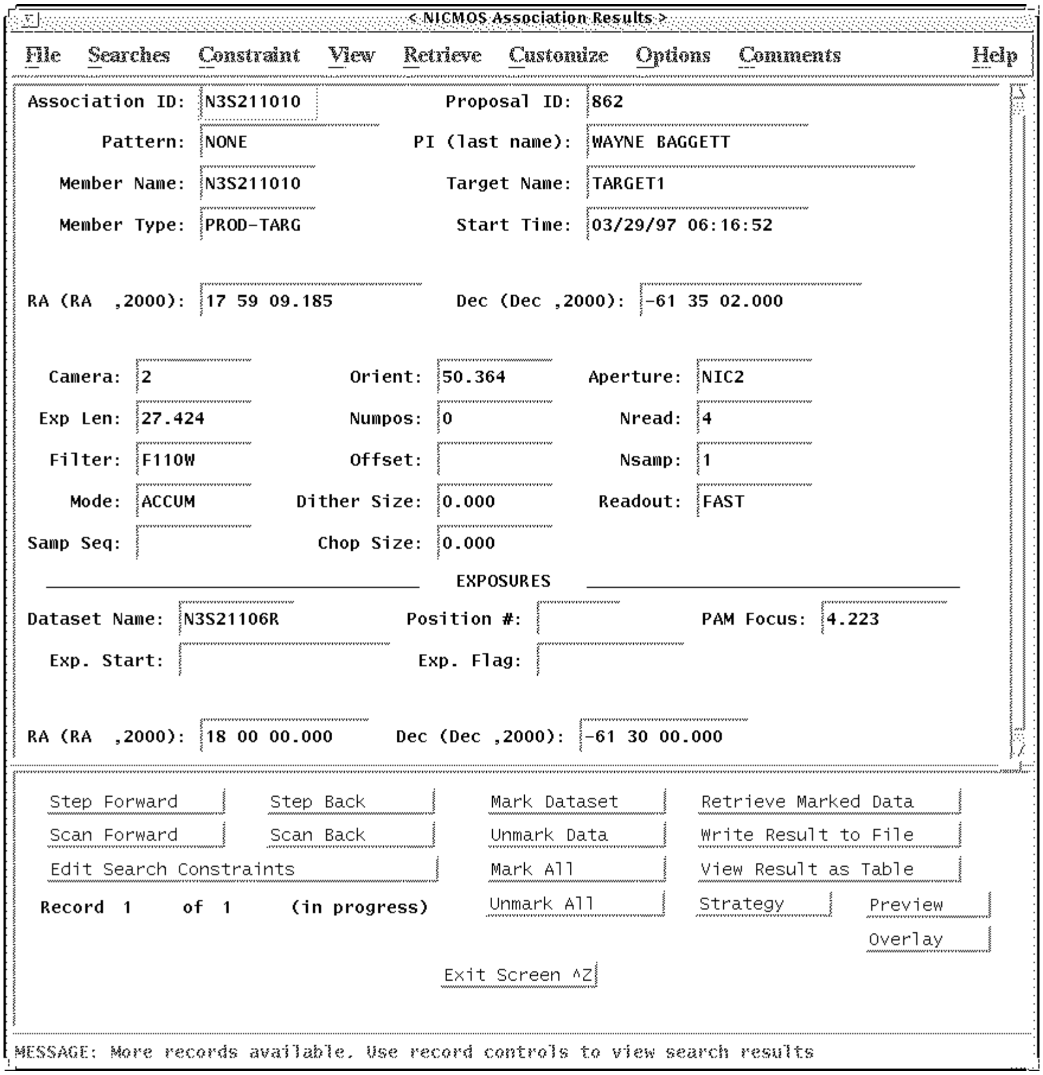
The STIS and NICMOS calibration pipelines sometimes produce single calibrated images from *associations* of many exposures. These associations allow HST pipeline processing to proceed further than it has in the past. For example, a NICMOS observer might specify a dithering pattern in a Phase II proposal. NICMOS would then take several exposures at offset positions, and the pipeline would combine them into a single mosaic. In this case, the original set of exposures constitutes the association, and the mosaic is the *association product*. Similarly, a STIS observer might specify a CR-SPLIT sequence in a Phase II proposal. STIS would gather several exposures at the same pointing, and the STIS pipeline would process this association of exposures into single image, free of cosmic rays, that would be the association product.

When you search the Archive with StarView for observations involving associations of exposures, your search will identify the final association product. The rootnames of association products always end in zero (see figure B.1 above.) If you request both Calibrated and Uncalibrated data from the Archive, you will receive both the association product and the exposures that went into making it. The corresponding association table, located in the file with suffix `asn` and the same rootname as the association product, lists the exposures belonging to the association. You can read this file using the STSDAS **tprint** or **tread** tasks (see table 3.1 in the HST Introduction). The exposure IDs in the association table share the same `ipppss` sequence as the association rootname, followed by a base 36 number `nn` (`n` = 0-9,A-Z) that uniquely identifies each exposure, and a character `t` that denotes the data transmission mode (see figure B.1).

In practice, STIS and NICMOS store the exposures belonging to associations differently. The exposures belonging to a STIS association all reside in the same file, while the exposures belonging to a NICMOS association reside in separate datasets. See the relevant Data Structures chapters for more details.

Information on the exposures belonging to an association is also available through StarView (see chapter 1 of the HST Introduction). From the <Welcome> Screen, click on **[HST Instrument Searches]** to get the <HST Instruments> screen, and then click on the **[Associations]** button for the instrument of interest. You can then search for the various exposures belonging to an association by entering the rootname of the association in the Association ID field and clicking on **[Begin Search]** . An Association Results Screen will display the results of the search, which you can step though using the **[Step Forward]** button. Figure B.1 below gives an example of a NICMOS Association Results Screen. Note the differences between the association rootname and coordinates and those of the individual exposure.

Figure B.1:Association Results Screen from StarView



Association ID: N3S211010 Proposal ID: 862

Pattern: NONE PI (last name): WAYNE BAGGETT

Member Name: N3S211010 Target Name: TARGET1

Member Type: PROD-TARG Start Time: 03/29/97 06:16:52

RA (RA ,2000): 17 59 09.185 Dec (Dec ,2000): -61 35 02.000

Camera: 2 Orient: 50.364 Aperture: NIC2

Exp Len: 27.424 Numpos: 0 Nread: 4

Filter: F110W Offset: Nsamp: 1

Mode: ACCUM Dither Size: 0.000 Readout: FAST

Samp Seq: Chop Size: 0.000

EXPOSURES

Dataset Name: N3S21106R Position #: PAM Focus: 4.223

Exp. Start: Exp. Flag:

RA (RA ,2000): 18 00 00.000 Dec (Dec ,2000): -61 30 00.000

Record 1 of 1 (in progress)

MESSAGE: More records available. Use record controls to view search results

APPENDIX C:

Observation Logs

In this appendix . . .

This Appendix describes the *Observation Log Files,* also known as OMS or *jitter* files. These files record pointing, jitter, and other Pointing Control System (PCS) data taken during an HST observation. You can use them to assess the behavior of the HST spacecraft during your observation, and in particular, to evaluate the jitter of the spacecraft while it was taking data. Here we describe the contents and structure of the observation log files, how to retrieve them from the Archive, and how to work with the data they contain.

C.1 Observation Log Files

Observation log files associated with each HST dataset contain pointing and specialized engineering data taken during the observation. These data files are produced by the Observatory Monitoring System (OMS), an automated software system that interrogates the HST engineering telemetry and correlates the time-tagged engineering stream with HST's Science Mission Schedule (SMS), the seven-day command and event list that drives all spacecraft activities. This system reports the status of the instruments and observatory and flags discrepancies between planned and executed actions. OMS provides observers with information about guide star acquisition, pointing, and tracking that is not normally provided in the science headers.

The observation log files share the same rootname as the observation they are associated with, except for the final character, which for observation log files is always a "`j`" (see appendix B for more on the names of HST data files). When OMS was installed in October 1994, it initially generated files with the suffixes `cmh`, `cmj`, `cmi`, which contained header information, high time resolution pointing data, and three-second average pointing data, respectively (see table C.1). OMS observation logs changed to the `jih/jid/jif` image format after August 1995, at which time the `cmi` table was renamed `jit` to keep the naming convention consistent. In the OMS version of August 1995, `cmj` tables were replaced with a jitter image, which is a two-dimensional histogram of jitter excursions during the observation. The suffixes of the GEIS jitter image are `jih` for the header and `jid` for the image data. The `jit` table accompanies the jitter image. The header file of the image replaces the `cmh` file but includes the same information with the addition of some image-related keywords.

A detailed description of the observation log files can be found on-line: http://www.stsci.edu/instruments/observatory/obslog/OL_1.html.

Table C.1: OMS Observation Log Files

Suffix	Contents
October 1994 to August 1995	
`cmh`	OMS header
`cmj`	High time resolution (IRAF table)
`cmi`	Three-second averages (IRAF table)
`_cmh.fits`	Archived FITS file of `cmh`
`_cmj.fits`	Archived FITS file of `cmj`
`_cmi.fits`	Archived FITS file of `cmi`
August 1995 to February 1997	
`jih/jid`	Two-dimensional histogram and header (GEIS)
`jit`	Three-second averages (IRAF table)[1]
`_jif.fits`	Archived FITS file which bundles the `jih/jid` files.
`_jit.fits`	Archived FITS file of `jit`.
February 1997 onward	
`_jif.fits`	Two-dimensional histogram (FITS)
`_jit.fits`	Three-second averages table (FITS)

[1] After May 11, 1995, the `jit` tables for exposures shorter than 6 seconds contain higher-resolution, one-second average pointing data.

Pointing and tracking information prior to October 1994 is not routinely available. Interested observers with data from this epoch, can send E-mail to help@stsci.edu.

C.1.1 Observation Log File Contents (October 1994 version)

Observation logs created between October 1994 and August 1995 contain:

- *rootname*`j.cmh`: This ASCII header file contains the time interval, the rootname, averages of the pointing and spacecraft jitter, the guiding mode, guide star information, and alert or failure keywords. Figure C.1 shows a representative observation log header file.
- *rootname*`j.cmj`: This table presents the data at the highest time resolution for the telemetry mode in use. It contains the reconstructed pointing, guide star coordinates, derived jitter at the instrument aperture, and guiding-related flags. The intent is: (1) to provide high-time resolution jitter data for deconvolution or for assessing small aperture pointing stability, and (2) to display the slew and tracking anomaly flags with the highest resolution. Table C.2 lists the table column heading, units and a brief definition.
- *rootname*`j.cmi`: This table contains data that were averaged over three-second intervals. It includes the same information as the `.cmj` table and also includes orbital data (e.g., latitude, longitude, limb angle, magnetic field values, etc.) and instrument-specific items. It is best suited for a quick-look assessment of pointing stability and for studying trends in telescope or instrument performance with orbital environment. Table C.3 lists the table column heading, units and a brief definition.
- rootnamej_cmi`/j/h.fits`: The above three GEIS files are actually archived as FITS files. They may be worked with as such, or run through the STSDAS task **strfits**, to convert them.
- Observation Log File Contents (August 1995 version)
- The contents of observation log files created between August 1995 and February 1997 are as follows:

- *rootname*`j.jih`: This GEIS header file, the analog to the `cmh` file, contains the time interval, the rootname, averages of the pointing and spacecraft jitter, the guiding mode, guide star information, and alert or failure keywords. Figure C.1 shows a representative observation log header file.
- *rootname*`j.jid`: This GEIS image—a significant enhancement of the old `cmj` file—presents a two-dimensional histogram of the pointing fluctuations during the observation. You can display it to visualize the spacecraft stability during you observation, and is information for deconvolutions and PSF analyses.
- *rootname*`j.jit`: This table, the analog to the `cmi` table, contains data that were averaged over three-second intervals. Its content is identical (see table C.3).
- *rootname*`j_jif.fits`: FITS file that is actually the de-archived product. This file can be converted to the `jih/jid` GEIS file via the **strftis** routine.
- *rootname*`j_jit.fits`: The de-archived FITS file corresponding to the `jit` IRAF table. It can be converted via **strfits**.

C.1.2 Observation Log File Contents (February 1997 version)

The contents of observation log files created since February 1997 are as follows:

- *rootname*`j_jif.fits`: The de-archived FITS file. Unlike the previous OMS epoch, this FITS file does not bundle a GEIS file and cannot be converted with **strfits**. This was done to more closely correlate the observation log files with the STIS and NICMOS FITS files with extensions and associations. OMS will normally put all associated observation logs into a single file, to correspond to the associated science exposures. However, if even one science exposure is orphaned (not associated) then an individual observation log FITS file will be produced for every exposure in that association. For a description of STIS and NICMOS association files, see appendix B. All of the information contained in the old `cmh/jih` ASCII header is now available as keywords in the FITS files.
- *rootname*`j_jit.fits`: The FITS file containing the table information. The comments for the `_jif` file apply here as well.

Table C.2: Contents of `.cmj` Table

Parameter	Units	Description
`seconds`	seconds	Time since window start
`V2 dom`	arcseconds	Dominant FGS V2 coordinate
`V3 dom`	arcseconds	Dominant FGS V3 coordinate
`V2 roll`	arcseconds	Roll FGS V2 coordinate
`V3 roll`	arcseconds	Roll FGS V3 coordinate
`SI V2`	arcseconds	Jitter at aperture reference
`SI V3`	arcseconds	Jitter at aperture reference
`RA`	degrees	Right ascension of aperture reference
`DEC`	degrees	Declination of aperture reference
`Roll`	degrees	Angle between North and +V3
`DayNight`	0,1 flag	Day (0) or night (1)
`Recenter`	0,1 flag	Recentering status
`TakeData`	0,1 flag	Vehicle guiding status
`SlewFlag`	0,1 flag	Vehicle slewing status

Figure C.1:A Representative .jih or .cmh Header

```
dSIMPLE  =                   F / data conforms to FITS standard                !
BITPIX  =                   32 / bits per data value                          !
DATATYPE= ‘INTEGER*4         ‘ / datatype of the group array                  !
NAXIS   =                    2 / number of data axes                          !
NAXIS1  =                   64 / length of the 1st data axis                  !
NAXIS2  =                   64 / length of the 2nd data axis                  !
GROUPS  =                    T / image is in group format                     !
GCOUNT  =                    1 / number of groups                             !
PCOUNT  =                    0 / number of parameters                         !
PSIZE   =                    0 / bits in the parameter block                  !
OMS_VER = ‘16.2C             ‘ / OMS version used to process this observation
PROCTIME= ‘1994.133:06:24:18.35’ / date-time OMS processed observation
                                 / date-times format (yyyy.ddd:hh:mm:ss.ss)

                               / IMAGE PARAMETERS
CRVAL1  =                  0.0 / right ascension of zero-jitter pixel (deg)
CRVAL2  =                  0.0 / declination of zero-jitter pixel (deg)
CRPIX1  =                   32 / x-coordinate of zero-jitter pixel
CRPIX2  =                   32 / y-coordinate of zero-jitter pixel
CTYPE1  = ‘RA---TAN          ‘ / first coordinate type
CTYPE2  = ‘DEC--TAN          ‘ / second coordinate type
CD1_1   =                  0.0 / partial of ra w.r.t. x (deg/pixel)
CD1_2   =                  0.0 / partial of ra w.r.t. y (deg/pixel)
CD2_1   =                  0.0 / partial of dec w.r.t. x (deg/pixel)
CD2_2   =                  0.0 / partial of dec w.r.t. y (deg/pixel)
COORDSYS= ‘WFPC2             ‘ / image coordinate system
XPIXINC =                  2.0 / plate scale along x (mas per pixel)
YPIXINC =                  2.0 / plate scale along y (mas per pixel)
PARITY  =                   -1 / parity between V2V3 frame and image frame
BETA1   =               134.72 / angle from +V3 to image +x (toward +V2)
BETA2   =               224.72 / angle from +V3 to image +y (toward +V2)

                               / OBSERVATION DATA
PROPOSID=                05233 / PEP proposal identifier
PROGRMID= ‘288               ‘ / program id (base 36)
OBSET_ID= ‘02                ‘ / observation set id
OBSERVTN= ‘03                ‘ / observation number (base 36)
TARGNAME= ‘NGC3379-PO        ‘ / proposer’s target name
STARTTIME= ‘1994.133:06:24:18.35’ / predicted observation window start time
ENDTIME = ‘1994.133:06:39:18.35’ / predicted observation window end time
SOGSID  = ‘U2880203          ‘ / SOGS observation name                        !

                               / SCIENTIFIC INSTRUMENT DATA
CONFIG  = ‘WFPC2             ‘ / proposed instrument configuration
PRIMARY = ‘SINGLE            ‘ / single, parallel-primary, parallel-secondary
OPERATE = ‘1994.133:06:22:46.91’ / predicted time instr. entered operate mode
TLMFORM = ‘PN                ‘ / telemetry format
APERTURE= ‘UWFALL            ‘ / aperture name
APER_V2 =                1.565 / V2 aperture position in vehicle frame (arcsec)
APER_V3 =                7.534 / V3 aperture position in vehicle frame (arcsec)

                               / SPACECRAFT DATA
ALTITUDE=               593.23 / average altitude during observation (km)
LOS_SUN =               106.08 / minimum line of sight to Sun (deg)
LOS_MOON=                77.11 / minimum line of sight to Moon (deg)
SHADOENT= ‘1994.133:05:11:29.00’ / predicted Earth shadow last entry
SHADOEXT= ‘1994.133:05:42:45.00’ / predicted Earth shadow last exit
LOS_SCV =                12.46 / minimum line of sight to S/C veloc. (deg)
LOS_LIMB=                 58.0 / average line of sight to Earth limb (deg)


                               / BACKGROUND LIGHT
ZODMOD  =                 22.3 / zodiacal light - model (V mag/arcsec2)
EARTHMOD=                 20.2 / peak Earth stray light - model (V mag/arcsec2)
MOONMOD =                 35.5 / moon stray light - model (V mag/arcsec2)
GALACTIC=                 -1.0 / diffuse galactic light - model (V mag/arcsec2)

                               / POINTING CONTROL DATA
GUIDECMD= ‘FINE LOCK         ‘ / commanded guiding mode
GUIDEACT= ‘FINE LOCK         ‘ / actual guiding mode at end of GS acquisition
GSD_ID  = ‘0084900235        ‘ / dominant guide star id
GSD_RA  =            161.70720 / dominant guide star RA (deg)
GSD_DEC =             12.45407 / dominant guide star DEC (deg)
```

Figure C.2:Representative `.jih` or `.cmh` Header

```
GSD_MAG =               12.867 / dominant guide star magnitude
GSR_ID  = ‘0085201189        ’ / roll guide star id
GSR_RA  =            161.93314 / roll guide star RA (deg)
GSR_DEC =             12.78141 / roll guide star DEC (deg)
GSR_MAG =               12.977 / roll guide star magnitude
GSACQ   = ‘1994.133:06:31:02.92’ / actual time of GS acquisition completion
PREDGSEP=             1420.775 / predicted guide star separation (arcsec)
ACTGSSEP=             1421.135 / actual guide star separation (arcsec)
GSSEPRMS=                  3.8 / RMS of guide star separation (milli-arcsec)
NLOSSES =                    0 / number of loss of lock events
LOCKLOSS=                  0.0 / total loss of lock time (sec)
NRECENT =                    0 / number of recentering events
RECENTR =                  0.0 / total recentering time (sec)

                               / LINE OF SIGHT JITTER SUMMARY
V2_RMS  =                  4.5 / V2 axis RMS (milli-arcsec)
V2_P2P  =                 51.6 / V2 axis peak to peak (milli-arcsec)
V3_RMS  =                 20.9 / V3 axis RMS (milli-arcsec)
V3_P2P  =                267.3 / V3 axis peak to peak (milli-arcsec)
RA_AVG  =            161.85226 / average RA (deg)
DEC_AVG =             12.58265 / average dec (deg)
ROLL_AVG=            293.01558 / average roll (deg)

                               / PROBLEM FLAGS, WARNINGS and STATUS MESSAGES
                               / (present only if problem exists)
ACQ2FAIL= ‘                 T’ / target acquisition failure
GSFAIL  = ‘DEGRADED          ’ / guide star acquisition failure  (*1)
TAPEDROP= ‘                 T’ / possible loss of science data
TLM_PROB= ‘                  ’ / problem with the engineering telemetry
TM_GAP  =               404.60 / duration of missing telemetry (sec)
SLEWING = ‘                 T’ / slewing occurred during this observation
TAKEDATA= ‘                 F’ / take data flag NOT on throughout observation
SIPROBnn= ‘                  ’ / problem with specified science instrument (*2)

END
-----------------------------------------------------------------------------

notes
*1 - GSFAIL appears only once in a single header file.
     The following table lists all current possible values for the GSFAIL
     keyword:
     -------------------------------------------------------------
     GSFAIL    |DEGRADED         | / guide star acquisition failure
               |IN PROGR         | / guide star acquisition failure
               |SSLEXP           | / guide star acquisition failure
               |SSLEXS           | / guide star acquisition failure
               |NOLOCK           | / guide star acquisition failure
               |                 |
               |SREXCS?          | / guide star acquisition failure
               |SREXCS1          | / guide star acquisition failure
               |SREXCS2          | / guide star acquisition failure
               |SREXCS3          | / guide star acquisition failure
               |                 |
               |SREXCP?          | / guide star acquisition failure
               |SREXCP1          | / guide star acquisition failure
               |SREXCP2          | / guide star acquisition failure
               |SREXCP3          | / guide star acquisition failure
               |                 |
               |UNKNOWN          | / guide star acquisition failure
               |VEHSAFE          | / guide star acquisition failure
     -------------------------------------------------------------

*2 - The SIPROBnn keywords appear in the header file with nn = 01 - 99.
     The following table lists all current possible values for the SIPROBnn
     keyword:
     -------------------------------------------------------------------------
     SIPROBnn  |DCF_NUM unchanged| / This observation may not have been taken
               |FOS Safing!      | / This observation affected when FOS Safed!
               |HRS Safing!      | / This observation affected when HRS Safed!
               |WFII Safing!     | / This observation affected when WFII Safed!
               |FOC Safing!      | / This observation affected when FOC Safed!
               |Shut             | / FOS aperture door is not Open!
               |FAILED           | / FGS astrometry target acquisition failed
     -------------------------------------------------------------------------
```

Table C.3: Contents of .jit or.cmiTable, Three-Second Averaging

Parameter	Units	Description
seconds	seconds	Time since window start
V2 dom	arcseconds	Dominant FGS V2 coordinate
V3 dom	arcseconds	Dominant FGS V3 coordinate
V2 roll	arcseconds	Roll FGS V2 coordinate
V3 roll	arcseconds	Roll FGS V3 coordinate
SI V2 AVG	arcseconds	Mean jitter in 3 seconds
SI V2 RMS	arcseconds	rms jitter in 3 seconds
SI V2 P2P	arcseconds	Peak jitter in 3 seconds
SI V3 AVG	arcseconds	Mean jitter in 3 seconds
SI V3 RMS	arcseconds	rms jitter in 3 seconds
SI V3 P2P	arcseconds	Peak jitter in 3 seconds
RA	degrees	Right ascension of aperture reference
DEC	degrees	Declination of aperture reference
Roll	degrees	Angle between North and +V3
LimbAng	degrees	Angle between earth limb and target
TermAng	degrees	Angle between terminator and target
LOS_Zenith	degrees	Angle between HST zenith and target
Latitude	degrees	HST subpoint latitude
Longitude	degrees	HST subpoint longitude
Mag V1,V2,V3	degrees	Magnetic field along V1, V2, V3
EarthMod	V Mag/arcsec^2	Model earth background light
SI_Specific	–	Special science instrument data
DayNight	0,1 flag	Day (0) or night (1)
Recenter	0,1 flag	Recentering status
TakeData	0,1 flag	Vehicle guiding status
SlewFlag	0,1 flag	Vehicle slewing status

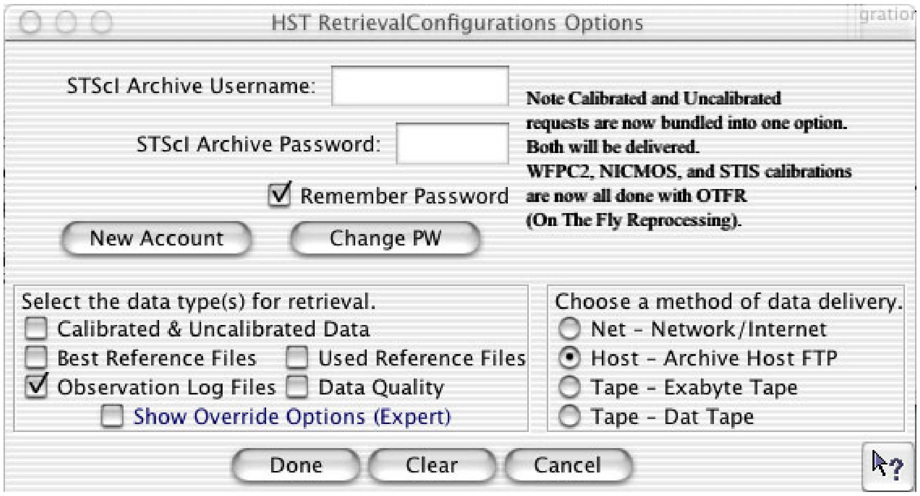
C.2 Retrieving Observation Logs

You can retrieve observation log files for data taken after October 20, 1994 from the HST Archive using StarView as described in chapter 1 of the HST Introduction. Unlike science data, which generally has a one-year proprietary period, observation log files become public as soon as they are archived.

The easiest way to get OMS files through StarView is to identify the observation of interest and proceed with your request as described in chapter 1 of the HST Introduction, until you reach the "HST Retrieval Configurations Options" screen, reproduced in figure C.3. You can then check the Observation Log Files box, along with any other desired boxes, and continue with your request. StarView will then deliver the associated observation log files.

For observations logged between October 1994 to August 1995, you will be delivered the `cmi`, `cmj`, and `cmh` files in FITS form (e.g., `_cmi.fits`). Observations archived from August 1995 to February 1997 will return `_jif.fits` and `_jit.fits` files. These, and the earlier FITS files can be worked with as such, or converted to their GEIS counterparts via the STSDAS **strfits** task. However, as of February 1997, the `_jif.fits` and `_jit.fits` files are standard FITS files with extensions and cannot be converted to GEIS.

Figure C.3:Choosing Observation Log Files in StarView



HST RetrievalConfigurations Options

STScI Archive Username:

STScI Archive Password:

☑ Remember Password

Note Calibrated and Uncalibrated requests are now bundled into one option. Both will be delivered.
WFPC2, NICMOS, and STIS calibrations are now all done with OTFR
(On The Fly Reprocessing).

Select the data type(s) for retrieval.
☐ Calibrated & Uncalibrated Data
☐ Best Reference Files ☐ Used Reference Files
☑ Observation Log Files ☐ Data Quality
☐ Show Override Options (Expert)

Choose a method of data delivery.
○ Net - Network/Internet
● Host - Archive Host FTP
○ Tape - Exabyte Tape
○ Tape - Dat Tape

C.3 Using Observation Logs

Here are some simple examples of what can be learned from the observation log files. Note that for FITS format observation logs, current versions of STSDAS tools will handle the files with extensions properly. Keywords can be viewed with tools such as **imheader** or **hedit**, and data viewed, plotted, or displayed using the same tasks one might have for the GEIS files. For more information on FITS file structures, see chapter 2 of the HST Introduction.

C.3.1 Guiding Mode

Unless requested, all observations will be scheduled with FINE LOCK guiding, which may be one or two guide stars (dominant and roll). The spacecraft may roll slightly during an observation if only one guide star is acquired. The amount of roll depends upon the gyro drift at the time of the observation, the location during an orbit, and the lever arm from the guide star to the center of the aperture.

There are three commanded guiding modes: FINE LOCK, FINE LOCK/GYRO, and GYRO. OMS header keywords GUIDECMD (commanded guiding mode) and GUIDEACT (actual guiding mode) will usually agree. If there was a problem, they won't agree and the GUIDEACT value will be the guiding method actually used during the exposure. If the acquisition of the second guide star fails, the spacecraft guidance, GUIDEACT, may drop from FINE LOCK to FINE LOCK/GYRO, or even to GYRO, which may result in a target rolling out of an aperture. Check the OMS header keywords to verify that there was no change in the requested guiding mode during the observation.

Until new flight software (version FSW 9.6) came online in September 1995, if the guide star acquisition failed, the guiding dropped to COARSE track. After September 1995, if the guide star acquisition failed, the tracking did not drop to COARSE track. Archival researchers may find older datasets that were obtained with COARSE track guiding.

The dominant and roll guide star keywords (GSD and GSR) in the OMS header can be checked to verify that two guide stars were used for guiding, or in the case of an acquisition failure, to identify the suspect guide star. The dominant and roll guide star keywords identify the stars that were scheduled to be used, and in the event of an acquisition failure, may not be the stars that were actually used. The following list of cmh keywords is an example of two star guiding.

```
GSD_ID  = ‘0853601369        ’ / Dominant Guide Star ID
GSD_RA  =           102.42595 / Dominant Guide Star RA (deg)
GSD_DEC =           -53.41362 / Dominant Guide Star DEC (deg)
GSD_MAG =              11.251 / Dominant Guide Star Magnitude
GSR_ID  = ‘0853602072        ’ / Roll Guide Star ID
GSR_RA  =           102.10903 / Roll Guide Star RA (deg)
GSR_DEC =           -53.77683 / Roll Guide Star DEC (deg)
GSR_MAG =              12.426 / Roll Guide Star Magnitude
```

If you suspect that a target has rolled out of the aperture during an exposure, you can quickly check the counts in each group of the raw science data. As an example, the following IRAF commands can be used to determine the counts in each group.

```
cl> grlist z2o4040dt.d0h 1-24 > groups.lis
cl> imstat @groups.lis
```

Some observations can span several orbits. If during a multiple orbit observation the guide star reacquisition fails, the observation may be terminated with possible loss of observing time, or switch to other less desirable guiding modes. The GSACQ keyword in the cmh header will state the time of the last successful guide star acquisition.

```
GSACQ   = ‘136:14:10:37.43   ’ / Actual time of GS Acquisition Completion
```

C.3.2 Guide Star Acquisition Failure

The guide star acquisition at the start of the observation set could fail if the FGS fails to lock onto the guide star. The target may not be in the aperture, or maybe only a piece of an extended target is in the aperture. The jitter values will be increased because FINE LOCK was not used. The following list of cmh header keywords indicate that the guide star acquisition failed.

```
V3_RMS =                   19.3 / V3 Axis RMS (milli-arcsec)
V3_P2P =                  135.7 / V3 Axis peak to peak (milli-arcsec)
GSFAIL = ‘          DEGRADED’ / Guide star acquisition failure!
```

The observation logs for all of the following observations in the observation set will have the “DEGRADED” guide star message. This is not a Loss of Lock situation but an actual failure to acquire the guide star in the desired guiding mode. For the example above, the guiding mode dropped from FINE LOCK to COARSE TRACK.

```
GUIDECMD= ‘FINE LOCK         ‘ / Commanded Guiding mode
GUIDEACT= ‘COARSE TRACK      ‘ / Actual Guiding mode at end of GS acquisition
```

If the observational dataset spans multiple orbits, the guide star will be re-acquired, but the guiding mode will not change from COARSE TRACK. In September 1995, the flight software was changed so that COARSE TRACK is no longer an option. The guiding mode drops from two guide star FINE LOCK to one guide star FINE LOCK , or to GYRO control.

C.3.3 Moving Targets and Spatial Scans

A type 51 slew is used to track moving targets (planets, satellites, asteroids, and comets). Observations are scheduled with FINE LOCK acquisition, i.e., with two or one guide stars. Usually, a guide star pair will stay within the pickle during the entire observation set, but if two guide stars are not available, a single guide star may be used, assuming the drift is small or the proposer says that the roll is not important for that particular observing program. An option during scheduling is to drop from FGS control to GYRO control when the guide stars move out of the FGS. Also, guide star handoffs (which are not a simple dropping of the guide stars to GYRO control) will affect the guiding and may be noticeable when the jitter ball is plotted.

The jitter statistics are accumulated at the start of the observation window. Moving targets and spatial scan motion will be seen in the jitter data and image. Therefore, the OMS header keywords V2_RMS and V3_RMS values (the root mean square of the jitter about the V2 and V3 axis) can be quite large for moving targets. Also, a special anomaly keyword (SLEWING) will be appended to the OMS header stating movement of the telescope during the observation. This is expected for observing moving targets. The following list of .cmh header keywords is an example of expected values while tracking a moving target.

```
                     / LINE OF SIGHT JITTER SUMMARY
V2_RMS  =        3.2 / V2 Axis RMS (milli-arcsec)
V2_P2P  =       17.3 / V2 Axis peak to peak (milli-arcsec)
V3_RMS  =       14.3 / V3 Axis RMS (milli-arcsec)
V3_P2P  =       53.6 / V3 Axis peak to peak (milli-arcsec)
RA_AVG  =  244.01757 / Average RA (deg)
DEC_AVG =  -20.63654 / Average DEC (deg)
ROLL_AVG=  280.52591 / Average Roll (deg)
SLEWING = ‘       T’ / Slewing occurred during this observation
```

C.3.4 High Jitter

The spacecraft may shake during an observation, even though the guiding mode is FINE LOCK. This movement may be due to a micro-meteorite hit, jitter at a day-night transition, or for some other unknown reasons. The FGS is quite stable and will track a guide star even during substantial spacecraft motion. The target may move about in an aperture, but the FGS will continue to track guide stars and reposition the target into the aperture. For most observations, the movement about the aperture during a spacecraft excursion will be quite small, but sometimes, especially for observations with the spectrographs, the aperture may move enough that the measured flux for the target will be less than a previous group. Check the OMS header keywords (V2_RMS, V3_RMS) for the root mean square of the jitter about the V2 and V3 axis. The following list of .cmh header keywords is an example of typical guiding rms values.

```
                               / LINE OF SIGHT JITTER SUMMARY
V2_RMS  =                  2.6 / V2 Axis RMS (milli-arcsec)
V2_P2P  =                 23.8 / V2 Axis peak to peak (milli-arcsec)
V3_RMS  =                  2.5 / V3 Axis RMS (milli-arcsec)
V3_P2P  =                 32.3 / V3 Axis peak to peak (milli-arcsec)
```

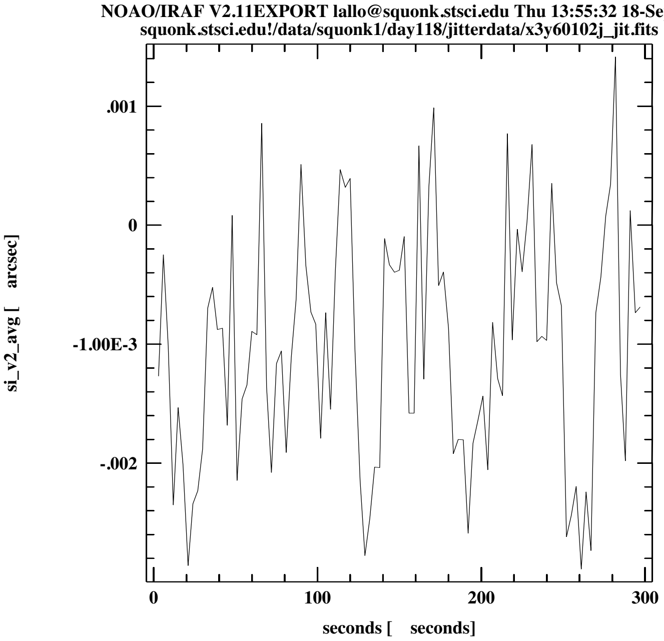
Recentering events occur when the spacecraft software decides that shaking is too severe to maintain lock. The FGS will release guide star control and within a few seconds reacquire the guide stars. It is assumed the guide stars are still within the FGS field of view. During the recentering time, INDEF will be written to the OMS table. Recentering events are tracked in the OMS header file.

Be careful when interpreting “Loss of Lock” and “Recentering” events that occur at the very beginning or at the end of the OMS window. The OMS window is larger than the observation window. These events might not affect the observation since the observation start time will occur after the guide stars are acquired (or re-acquired), and the observation stop time may occur before the “Loss of Lock” or “Recentering” event that occurred at the end of an OMS window.

The **sgraph** commend in the **stsdas.graphics.stplot** package will plot time vs. jitter along the direction of HST’s V2 axis (see figure C.4):

```
cl> sgraph "x3y60102j_jit.fits seconds si_v2_avg"
```

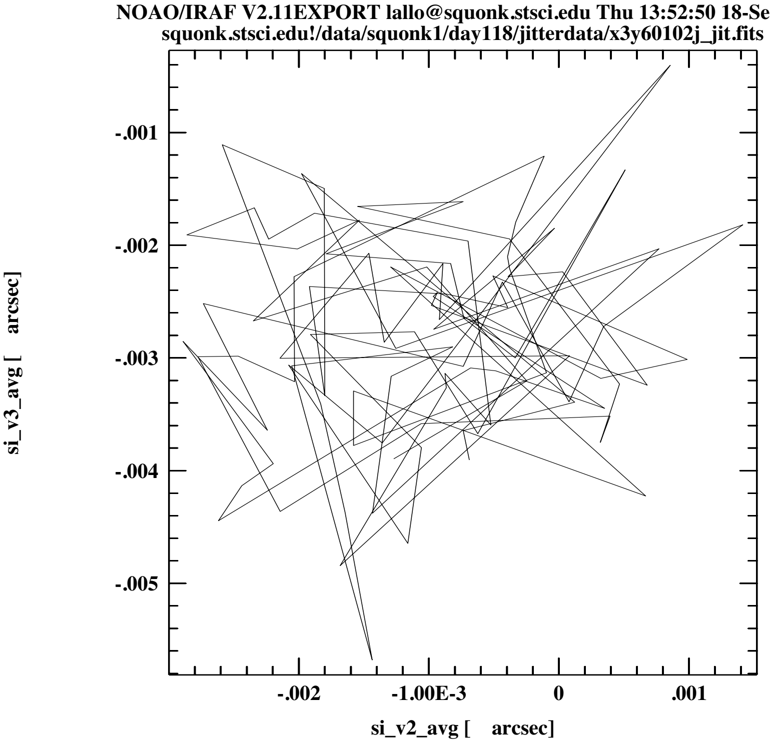
Figure C.4:Plotting Jitter Along V3 Axis



To get an idea of pointing stability, you can create a *jitter ball* by plotting jitter along the V2 axis vs. jitter along the V3 axis (see figure C.5):

```
st> sgraph "x3660102j_jit.fits si_v2_avg si_v3_avg"
```

Figure C.5:Plotting V2 vs. V3 Jitter



The **tstatistics** task can be used to find the mean value of the `si_v3_avg` column—the amount of jitter (in arcseconds) in the direction of the V3. This value can be used to model jitter in a PSF. In this example, the mean jitter is ~3 mas, which is typical for post-servicing mission data:

Figure C.6:Averaging a Column with tstatistics

```
tt> tstat u26m0801j.cmi si_v3_avg
# u26m0801j.cmi  si_v3_avg
#
nrows            mean       stddev       median           min         max
   11   -0.003006443888  0.00362533  -7.17163E-4  -0.00929515  0.00470988
```

Understanding and interpreting the meaning of the table columns and header keywords is critical to understanding the observation logs. Please read the available documentation and contact the STScI Help Desk (help@stsci.edu.) if you have any questions about the files. Documentation is available via the WWW at: http://www.stsci.edu/instruments/observatory/obslog/OL_1.html.

Index